

# Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/000063

International filing date: 06 January 2005 (06.01.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP  
Number: 2004-000752  
Filing date: 06 January 2004 (06.01.2004)

Date of receipt at the International Bureau: 04 February 2005 (04.02.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland  
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

07.01.2005

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日            2 0 0 4 年   1 月   6 日  
Date of Application:

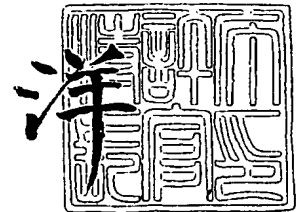
出 願 番 号            特 願 2 0 0 4 - 0 0 0 7 5 2  
Application Number:  
[ST. 10/C]:            [ J P 2 0 0 4 - 0 0 0 7 5 2 ]

出   願   人            ソニー株式会社  
Applicant(s):

2 0 0 4 年 1 1 月 1 1 日

特許庁長官  
Commissioner,  
Japan Patent Office

小 川



【書類名】 特許願  
【整理番号】 0390566204  
【提出日】 平成16年 1月 6日  
【あて先】 特許庁長官殿  
【国際特許分類】 G06T 5/00  
【発明者】  
    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内  
    【氏名】 近藤 哲二郎  
【発明者】  
    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内  
    【氏名】 渡辺 勉  
【特許出願人】  
    【識別番号】 000002185  
    【氏名又は名称】 ソニー株式会社  
【代理人】  
    【識別番号】 100082131  
    【弁理士】  
    【氏名又は名称】 稲本 義雄  
    【電話番号】 03-3369-6479  
【手数料の表示】  
    【予納台帳番号】 032089  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 9708842

**【書類名】 特許請求の範囲****【請求項 1】**

移動するオブジェクトを明確に表示する画像処理装置であって、  
画像の中の移動するオブジェクトの特徴点を検出する特徴点検出手段と、  
前記特徴点検出手段により検出された前記特徴点に基づいて、前記画像の中の前記オブジェクトの移動を追尾する追尾手段と、  
前記追尾手段による追尾結果に基づいて、前記画像の中の前記オブジェクトの周囲に予め設定された大きさの補正領域を設定する補正領域設定手段と、  
前記画像の中の前記補正領域内の画像を補正する補正手段と、  
前記補正領域内が前記補正手段により補正された画像の表示を制御する表示制御手段とを備えることを特徴とする画像処理装置。

**【請求項 2】**

前記補正手段は、前記画像のぼけを補正することを特徴とする請求項 1 に記載の画像処理装置。

**【請求項 3】**

前記補正手段は、  
前記補正領域内の画像を特定する制御信号と、画像のぼけの度合いを表すパラメータを供給する供給手段と、  
前記制御信号に基づいて特定された前記補正領域内の画像の特徴を検出し、検出された特徴を表す特徴コードを出力する特徴検出手段と、  
前記画像のぼけの度合いを表すパラメータと、前記特徴検出手段により出力された特徴コードに対応する係数を記憶する記憶手段と、  
前記記憶手段から、前記パラメータと前記特徴検出手段により出力された前記特徴コードに対応する係数を読み出す読み出し手段と、  
前記読み出し手段により読み出された係数に基づいて、入力画像の画素の値に対して積和演算を行う積和演算手段と、  
前記積和演算手段による演算結果と前記入力画像の画素の値を選択して出力する選択出力手段とを備え、  
前記補正領域内の画像のぼけを除去するように補正することを特徴とする請求項 2 に記載の画像処理装置。

**【請求項 4】**

前記特徴検出手段は、  
入力画像の中から、積和演算を行う画素の周囲の、予め設定された第 1 の領域に含まれる複数の画素を抽出する第 1 の抽出手段と、  
前記第 1 の領域と、垂直または水平の複数の方向に連続した、複数の第 2 の領域に含まれる複数の画素を抽出する第 2 の抽出手段と、  
前記第 1 の抽出手段により抽出された画素と、前記第 2 の抽出手段により抽出された画素において、対応する画素の値の差分の絶対値の総和を求めて、複数のブロック差分を演算するブロック差分演算手段と、  
前記ブロック差分が、予め設定された閾値より大きいかな否かを判定する差分判定手段とを備えることを特徴とする請求項 3 に記載の画像処理装置。

**【請求項 5】**

前記パラメータは、ぼけた画像の画素とぼけていない画像の画素の関係を表すモデル式におけるガウス関数のパラメータであることを特徴とする請求項 3 に記載の画像処理装置。

**【請求項 6】**

前記記憶手段により記憶される係数は、前記モデル式の逆行列を演算することにより求められた係数であることを特徴とする請求項 5 に記載の画像処理装置。

**【請求項 7】**

前記選択出力手段は、

前記積和演算手段により積和演算が行われた複数の画素を抽出する第 1 の抽出手段と

、  
前記第 1 の抽出手段により抽出された複数の画素の分散の度合いを表す分散度を演算する分散演算手段と、

前記分散演算手段により演算された分散度が予め設定された閾値より大きいか否かを判定する分散判定手段と

を備えることを特徴とする請求項 3 に記載の画像処理装置。

**【請求項 8】**

前記選択出力手段は、前記分散判定手段の判定結果に基づいて、出力する画素の値を、前記積和演算手段による演算結果、または前記入力画像の画素の値のいずれかから選択する画素選択手段

をさらに備えることを特徴とする請求項 7 に記載の画像信号処理装置。

**【請求項 9】**

画像の中で、移動するオブジェクトを明確に表示する画像処理装置の画像処理方法であって、

画像の中の移動するオブジェクトの特徴点を検出する特徴点検出ステップと、

前記特徴点検出ステップの処理により検出された前記特徴点に基づいて、画像の前記オブジェクトの移動を追尾する追尾ステップと、

前記追尾ステップの処理による追尾結果に基づいて、前記画像の中に前記オブジェクトの周囲に予め設定された大きさの補正領域を設定する補正領域設定ステップと、

前記画像の中の前記補正領域内の画像を補正する補正ステップと、

前記補正領域内が前記補正ステップの処理により補正された画像の表示を制御する表示制御ステップと

を含むことを特徴とする画像処理方法。

**【請求項 10】**

画像の中で、移動するオブジェクトを明確に表示する画像処理装置のプログラムであって、

画像の中の移動するオブジェクトの特徴点の検出を制御する特徴点検出制御ステップと

、  
前記特徴点検出制御ステップの処理により検出された前記特徴点に基づいて、画像の前記オブジェクトの移動を追尾するように制御する追尾制御ステップと、

前記追尾制御ステップの処理による追尾結果に基づいて、前記画像の中に前記オブジェクトの周囲に予め設定された大きさの補正領域を設定するように制御する補正領域設定制御ステップと、

前記画像の中の前記補正領域内の画像の補正を制御する補正制御ステップと、

前記補正領域内が前記補正制御ステップの処理により補正された画像の表示を制御する表示制御ステップと

をコンピュータに実行させることを特徴とするプログラム。

**【請求項 11】**

画像の中で、移動するオブジェクトを明確に表示する画像処理装置のプログラムが記録される記録媒体であって、

画像の中の移動するオブジェクトの特徴点の検出を制御する特徴点検出制御ステップと

、  
前記特徴点検出制御ステップの処理により検出された前記特徴点に基づいて、画像の前記オブジェクトの移動を追尾するように制御する追尾制御ステップと、

前記追尾制御ステップの処理による追尾結果に基づいて、前記画像の中に前記オブジェクトの周囲に予め設定された大きさの補正領域を設定するように制御する補正領域設定制御ステップと、

前記画像の中の前記補正領域内の画像の補正を制御する補正制御ステップと、  
前記補正領域内が前記補正制御ステップの処理により補正された画像の表示を制御する  
表示制御ステップと  
をコンピュータに実行させるプログラムが記録されることを特徴とする記録媒体。

【書類名】明細書

【発明の名称】画像処理装置および方法、プログラム並びに記録媒体

【技術分野】

【0001】

本発明は、画像処理装置および方法、プログラム並びに記録媒体に関し、特に、画像の中で移動するオブジェクトを発見しやすく表示できるようにする画像処理装置および方法、プログラム並びに記録媒体に関する。

【背景技術】

【0002】

ホームセキュリティシステムでは、撮像装置から送信される監視画像がモニタTV (Television)に表示される。このようなシステムにおいて、マイクロ波センサと画像センサを組み合わせる監視装置を構成し、侵入者の監視の精度を向上させる方法が提案されている(例えば、特許文献1参照)。

【0003】

また、動画像として表示される画像の中で、移動する(動く)物体を追尾対象とし、その追尾点を、自動的に追尾して画像を表示する方法も提案されている(例えば、特許文献2参照)。

【特許文献1】特開平11-161860号公報

【特許文献2】特開平6-143235号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

しかしながら、従来の技術では、例えば、不正な侵入者など、画像の中で動く物体(オブジェクト)を検知して、追尾しても、ユーザに対して表示される画像の中で、どの部分が、注目すべきオブジェクトであるのかを明確に表示することができず、その結果、ユーザが注目すべきオブジェクトを発見するのに時間がかかるおそれがあるという課題があった。

【0005】

本発明はこのような状況に鑑みてなされたものであり、画像の中で移動するオブジェクトを発見しやすく表示できるようにするものである。

【課題を解決するための手段】

【0006】

本発明の画像処理装置は、移動するオブジェクトを明確に表示する画像処理装置であって、画像の中の移動するオブジェクトの特徴点を検出する特徴点検出手段と、特徴点検出手段により検出された特徴点に基づいて、画像の中のオブジェクトの移動を追尾する追尾手段と、追尾手段による追尾結果に基づいて、画像の中のオブジェクトの周囲に予め設定された大きさの補正領域を設定する補正領域設定手段と、画像の中の補正領域内の画像を補正する補正手段と、補正領域内が補正手段により補正された画像の表示を制御する表示制御手段とを備えることを特徴とする。

【0007】

前記補正手段は、画像のぼけを補正するようにすることができる。

【0008】

前記補正手段は、補正領域内の画像を特定する制御信号と、画像のぼけの度合いを表すパラメータを供給する供給手段と、制御信号に基づいて特定された補正領域内の画像の特徴を検出し、検出された特徴を表す特徴コードを出力する特徴検出手段と、画像のぼけの度合いを表すパラメータと、特徴検出手段により出力された特徴コードに対応する係数を記憶する記憶手段と、記憶手段から、パラメータと特徴検出手段により出力された特徴コードに対応する係数を読み出す読み出し手段と、読み出し手段により読み出された係数に基づいて、入力画像の画素の値に対して積和演算を行う積和演算手段と、積和演算手段による演算結果と入力画像の画素の値を選択して出力する選択出力手段とを備え、補正領域内

の画像のぼけを除去するように補正するようにすることができる。

【0009】

前記特徴検出手段は、入力画像の中から、積和演算を行う画素の周囲の、予め設定された第1の領域に含まれる複数の画素を抽出する第1の抽出手段と、第1の領域と、垂直または水平の複数の方向に連続した、複数の第2の領域に含まれる複数の画素を抽出する第2の抽出手段と、第1の抽出手段により抽出された画素と、第2の抽出手段により抽出された画素において、対応する画素の値の差分の絶対値の総和を求めて、複数のブロック差分を演算するブロック差分演算手段と、ブロック差分が、予め設定された閾値より大きいかなかを判定する差分判定手段とを備えるようにすることができる。

【0010】

前記パラメータは、ぼけた画像の画素とぼけていない画像の画素の関係を表すモデル式におけるガウス関数のパラメータであるようにすることができる。

【0011】

前記記憶手段により記憶される係数は、モデル式の逆行列を演算することにより求められた係数であるようにすることができる。

【0012】

前記選択出力手段は、積和演算手段により積和演算が行われた複数の画素を抽出する第1の抽出手段と、第1の抽出手段により抽出された複数の画素の分散の度合いを表す分散度を演算する分散演算手段と、分散演算手段により演算された分散度が予め設定された閾値より大きいかなかを判定する分散判定手段とを備えるようにすることができる。

【0013】

前記選択出力手段は、分散判定手段の判定結果に基づいて、出力する画素の値を、積和演算手段による演算結果、または入力画像の画素の値のいずれかから選択する画素選択手段をさらに備えるようにすることができる。

【0014】

本発明の画像処理方法は、画像の中で、移動するオブジェクトを明確に表示する画像処理装置の画像処理方法であって、画像の中の移動するオブジェクトの特徴点を検出する特徴点検出ステップと、特徴点検出ステップの処理により検出された特徴点に基づいて、画像のオブジェクトの移動を追尾する追尾ステップと、追尾ステップの処理による追尾結果に基づいて、画像の中にオブジェクトの周囲に予め設定された大きさの補正領域を設定する補正領域設定ステップと、画像の中の補正領域内の画像を補正する補正ステップと、補正領域内が補正ステップの処理により補正された画像の表示を制御する表示制御ステップとを含むことを特徴とする。

【0015】

本発明のプログラムは、画像の中で、移動するオブジェクトを明確に表示する画像処理装置のプログラムであって、画像の中の移動するオブジェクトの特徴点の検出を制御する特徴点検出制御ステップと、特徴点検出制御ステップの処理により検出された特徴点に基づいて、画像のオブジェクトの移動を追尾するように制御する追尾制御ステップと、追尾制御ステップの処理による追尾結果に基づいて、画像の中にオブジェクトの周囲に予め設定された大きさの補正領域を設定するように制御する補正領域設定制御ステップと、画像の中の補正領域内の画像の補正を制御する補正制御ステップと、補正領域内が補正制御ステップの処理により補正された画像の表示を制御する表示制御ステップとをコンピュータに実行させることを特徴とする。

【0016】

本発明の記録媒体は、画像の中で、移動するオブジェクトを明確に表示する画像処理装置のプログラムが記録される記録媒体であって、画像の中の移動するオブジェクトの特徴点の検出を制御する特徴点検出制御ステップと、特徴点検出制御ステップの処理により検出された特徴点に基づいて、画像のオブジェクトの移動を追尾するように制御する追尾制御ステップと、追尾制御ステップの処理による追尾結果に基づいて、画像の中にオブジェクトの周囲に予め設定された大きさの補正領域を設定するように制御する補正領域設定制



御ステップと、画像の中の補正領域内の画像の補正を制御する補正制御ステップと、補正領域内が前記補正制御ステップの処理により補正された画像の表示を制御する表示制御ステップとをコンピュータに実行させるプログラムが記録されることを特徴とする。

【0017】

本発明の画像処理装置および方法、並びにプログラムにおいては、画像の中の移動するオブジェクトの特徴点が検出され、検出された特徴点に基づいて、画像のオブジェクトの移動が追尾され、その追尾結果に基づいて、画像の中にオブジェクトの周囲に予め設定された大きさの補正領域が設定され、画像の中の補正領域内の画像が補正され、補正領域内が補正された画像の表示が制御される。

【発明の効果】

【0018】

本発明によれば、注目すべきオブジェクトを発見しやすくすることができる。特に、画像の中で移動するオブジェクトを発見しやすく表示することができる。

【発明を実施するための最良の形態】

【0019】

以下に本発明の実施の形態を説明するが、本明細書に記載した発明と、発明の実施の形態との対応関係を例示すると、次のようになる。この記載は、本明細書に記載されている発明をサポートする実施の形態が明細書に記載されていることを確認するためのものである。従って、明細書には記載されているが、ここには記載されていない実施の形態があったとしても、そのことは、その実施の形態が、その発明に対応するものではないことを意味するものではない。逆に、実施の形態が発明に対応するものとしてここに記載されていたとしても、そのことは、その実施の形態が、その発明以外の発明には対応しないものであることを意味するものでもない。

【0020】

さらに、この記載は、明細書に記載されている発明が、全て請求されていることを意味するものではない。換言すれば、この記載は、明細書に記載されている発明であって、この出願では請求されていない発明の存在、すなわち、将来、分割出願されたり、補正により出願、または追加される発明の存在を否定するものではない。

【0021】

請求項1の画像処理装置は、移動するオブジェクトを明確に表示する画像処理装置（例えば、図1の監視カメラシステム1）であって、画像の中の移動するオブジェクト（例えば、図3のオブジェクト51）の特徴点（例えば、図3の追尾点51A）を検出する特徴点検出手段（例えば、図1の追尾対象検出部24）と、前記特徴点検出手段により検出された前記特徴点に基づいて、前記画像の中の前記オブジェクトの移動を追尾する追尾手段（例えば、図1の追尾対象検出部24）と、前記追尾手段による追尾結果に基づいて、前記画像の中の前記オブジェクトの周囲に予め設定された大きさの補正領域（例えば、図3の補正対象エリア52）を設定する補正領域設定手段（例えば、図1のエリア設定部25）と、前記画像の中の前記補正領域内の画像を補正する補正手段（例えば、図1の画像補正部22）と、前記補正領域内が前記補正手段により補正された画像の表示を制御する表示制御手段（例えば、図1の画像ディスプレイ23）とを備える。

【0022】

請求項2の画像処理装置は、前記補正手段が、前記画像のぼけ（例えば、フォーカスぼけ）を補正するようにすることができる。

【0023】

請求項3の画像処理装置は、前記補正手段が、前記補正領域内の画像を特定する制御信号と、画像のぼけの度合いを表すパラメータ（例えば、パラメータ $\sigma$ ）を供給する供給手段（例えば、図49の制御信号生成部741）と、前記制御信号に基づいて特定された前記補正領域内の画像の特徴を検出し、検出された特徴を表す特徴コードを出力する特徴検出手段（例えば、図49の画像特徴検出部742）と、前記画像のぼけの度合いを表すパラメータと、前記特徴検出手段により出力された特徴コードに対応する係数を記憶する記憶

手段（例えば、図49の係数ROM744）と、前記記憶手段から、前記パラメータと前記特徴検出手段により出力された前記特徴コードに対応する係数を読み出す読み出し手段（例えば、図49のアドレス演算部743）と、前記読み出し手段により読み出された係数に基づいて、入力画像の画素の値に対して積和演算を行う積和演算手段（例えば、図49の積和演算部746）と、前記積和演算手段による演算結果と前記入力画像の画素の値を選択して出力する選択出力手段（例えば、図49の画像合成部747）とを備え、前記補正領域内の画像のぼけを除去するように補正することができる。

#### 【0024】

請求項4の画像処理装置は、前記特徴検出手段が、入力画像の中から、積和演算を行う画素の周囲の、予め設定された第1の領域に含まれる複数の画素を抽出する第1の抽出手段（例えば、図60のブロック切り出し部841-5）と、前記第1の領域と、垂直または水平の複数の方向に連続した、複数の第2の領域に含まれる複数の画素を抽出する第2の抽出手段（例えば、図60のブロック切り出し部841-1乃至841-4）と、前記第1の抽出手段により抽出された画素と、前記第2の抽出手段により抽出された画素において、対応する画素の値の差分の絶対値の総和を求めて、複数のブロック差分を演算するブロック差分演算手段（例えば、図60のブロック差分演算部842-1乃至842-4）と、前記ブロック差分が、予め設定された閾値より大きいかなんかを判定する差分判定手段（例えば、図60の閾値判定部843-1乃至843-2）とを備えるようにすることができる。

#### 【0025】

請求項5の画像処理装置は、前記パラメータが、ぼけた画像の画素とぼけていない画像の画素の関係を表すモデル式（例えば、式（6））におけるガウス関数（例えば、式（7））のパラメータであるようにすることができる。

#### 【0026】

請求項7の画像処理装置は、前記選択出力手段が、前記積和演算手段により積和演算が行われた複数の画素を抽出する第1の抽出手段（例えば、図63のブロック切り出し部901）と、前記第1の抽出手段により抽出された複数の画素の分散の度合いを表す分散度を演算する分散演算手段（例えば、図63の分散演算部902）と、前記分散演算手段により演算された分散度が予め設定された閾値より大きいかなんかを判定する分散判定手段（例えば、図63の閾値判定部903）とを備えるようにすることができる。

#### 【0027】

請求項8の画像処理装置は、前記選択出力手段が、前記分散判定手段の判定結果に基づいて、出力する画素の値を、前記積和演算手段による演算結果、または前記入力画像の画素の値のいずれかから選択する画素選択手段（例えば、図63の選択部904）をさらに備えるようにすることができる。

#### 【0028】

請求項9の画像処理方法は、画像の中で、移動するオブジェクトを明確に表示する画像処理装置（例えば、図1の監視カメラシステム1）の画像処理方法であって、画像の中の移動するオブジェクト（例えば、図3のオブジェクト51）の特徴点（例えば、図3の追尾点51A）を検出する特徴点検出ステップ（例えば、図2のステップS1）と、前記特徴点検出ステップの処理により検出された前記特徴点に基づいて、画像の前記オブジェクトの移動を追尾する追尾ステップ（例えば、図2のステップS2）と、前記追尾ステップの処理による追尾結果に基づいて、前記画像の中に前記オブジェクトの周囲に予め設定された大きさの補正領域（例えば、図3の補正対象エリア52）を設定する補正領域設定ステップ（例えば、図2のステップS3）と、前記画像の中の前記補正領域内の画像を補正する補正ステップ（例えば、図2のステップS4）と、前記補正領域内が前記補正ステップの処理により補正された画像の表示を制御する表示制御ステップ（例えば、図2のステップS5）とを含む。

#### 【0029】

請求項10のプログラムは、画像の中で、移動するオブジェクトを明確に表示する画像

処理装置（例えば、図1の監視カメラシステム1）のプログラムであって、画像の中の移動するオブジェクト（例えば、図3のオブジェクト51）の特徴点（例えば、図3の追尾点51A）の検出を制御する特徴点検出制御ステップ（例えば、図2のステップS1）と、前記特徴点検出制御ステップの処理により検出された前記特徴点に基づいて、画像の前記オブジェクトの移動を追尾するように制御する追尾制御ステップ（例えば、図2のステップS2）と、前記追尾制御ステップの処理による追尾結果に基づいて、前記画像の中に前記オブジェクトの周囲に予め設定された大きさの補正領域（例えば、図3の補正対象エリア52）を設定するように制御する補正領域設定制御ステップ（例えば、図2のステップS3）と、前記画像の中の前記補正領域内の画像の補正を制御する補正制御ステップ（例えば、図2のステップS4）と、前記補正領域内が前記補正制御ステップの処理により補正された画像の表示を制御する表示制御ステップ（例えば、図2のステップS5）とをコンピュータに実行させる。

#### 【0030】

請求項11の記録媒体は、画像の中で、移動するオブジェクトを明確に表示する画像処理装置（例えば、図1の監視カメラシステム1）のプログラムが記録される記録媒体であって、画像の中の移動するオブジェクト（例えば、図3のオブジェクト51）の特徴点（例えば、図3の追尾点51A）の検出を制御する特徴点検出制御ステップ（例えば、図2のステップS1）と、前記特徴点検出制御ステップの処理により検出された前記特徴点に基づいて、画像の前記オブジェクトの移動を追尾するように制御する追尾制御ステップ（例えば、図2のステップS2）と、前記追尾制御ステップの処理による追尾結果に基づいて、前記画像の中に前記オブジェクトの周囲に予め設定された大きさの補正領域（例えば、図3の補正対象エリア52）を設定するように制御する補正領域設定制御ステップ（例えば、図2のステップS3）と、前記画像の中の前記補正領域内の画像の補正を制御する補正制御ステップ（例えば、図2のステップS4）と、前記補正領域内が前記補正制御ステップの処理により補正された画像の表示を制御する表示制御ステップ（例えば、図2のステップS5）とをコンピュータに実行させるプログラムが記録される。

#### 【0031】

以下、図面を参照して、本発明の実施の形態について説明する。図1は、本発明を監視カメラシステムに適用した場合の構成例を表している。この監視カメラシステム1においては、CCDビデオカメラ等よりなる撮像部21により撮像された画像が画像ディスプレイ23に表示される。追尾対象検出部24は、撮像部21より入力された画像から追尾対象を検出し、検出結果をオブジェクト追尾部26に出力する。

#### 【0032】

オブジェクト追尾部26は、撮像部21より供給された画像中の、追尾対象検出部24により指定された追尾点を追尾するように動作する。エリア設定部25は、オブジェクト追尾部26からの出力結果に基づいて、撮像部21により撮像された画像の中から、追尾点を含むオブジェクトの周辺の所定の領域（エリア）を設定し、その領域を指定する位置情報を画像補正部22に出力する。画像補正部22は、撮像部21により撮像された画像の中のエリア設定部25により設定された領域について、画像のぼけ（フォーカスぼけ）を除去する補正を行い、画像ディスプレイ23に出力する。カメラ駆動部27は、オブジェクト追尾部26からの制御に基づいて、撮像部21が追尾点を中心とする画像を撮影するように撮像部21を駆動する。

#### 【0033】

制御部27は、例えば、マイクロコンピュータなどにより構成され、各部を制御する。制御部27には、半導体メモリ、磁気ディスク、光ディスク、光磁気ディスクなどにより構成されるリムーバブルメディア28が必要に応じて接続され、プログラム、その他各種のデータが必要に応じて供給される。制御部27は、また、図示せぬ入出力インタフェースを介して、ユーザからの指示（コマンドなど）の入力を受け付ける。

#### 【0034】

次に、図2のフローチャートを参照して、監視処理の動作について説明する。監視シス

テム1の電源がオンされているとき、撮像部21は監視する領域を撮像し、その撮像して得られた画像を追尾対象検出部24、オブジェクト追尾部26、および画像補正部22を介して画像ディスプレイ23に出力する。追尾対象検出部24は、ステップS1において、撮像部21から入力された画像から追尾対象を検出する処理を実行する。追尾対象検出部24は、例えば、動く物体が検出された場合、その動く物体を追尾対象として検出する。追尾対象検出部24は、追尾対象の中から、例えば、最も輝度の高い点、あるいは追尾対象の中心の点などを追尾点として検出し、オブジェクト追尾部26に出力する。

#### 【0035】

ステップS2において、オブジェクト追尾部26は、ステップS1で検出された追尾点を追尾する追尾処理を実行する。追尾処理の詳細については、図6を参照して後述するが、この処理により、撮像部21により撮像された画像の中の追尾対象となるオブジェクト（例えば、人、動物など）の中の追尾点（例えば、目、頭の中心）が追尾され、その追尾位置を表す位置情報がエリア設定部25に出力される。

#### 【0036】

ステップS3において、エリア設定部25は、オブジェクト追尾部26からの出力に基づいて、追尾対象のオブジェクトの周辺の所定の領域（例えば、追尾点を中心として、所定の大きさの四角形で表される領域）を補正対象エリアとして設定する。

#### 【0037】

ステップS4において、画像補正部22は、撮像部21により撮像された画像の中で、エリア設定部25により設定された補正対象エリア内の画像を補正する画像補正処理を実行する。画像補正処理の詳細については、図58を参照して後述するが、この処理により、補正対象エリア内の画像について、画像のぼけが除去された鮮明な映像が提供される。

#### 【0038】

ステップS5において、画像ディスプレイ23は、ステップS4で補正された画像、すなわち、撮像部21により撮像された画像であって、特に、補正対象エリア内のみが鮮明になるように補正された画像を出力する。

#### 【0039】

ステップS6において、オブジェクト追尾部26は、ステップS2の処理による追尾結果に基づいて、オブジェクトの移動を検出し、移動したオブジェクトを撮像できるようにカメラを駆動させるカメラ駆動信号を生成し、カメラ駆動部27に出力する。ステップS7において、カメラ駆動部27はカメラ駆動信号に基づいて撮像部21を駆動する。これにより撮像部21は、追尾点が画面のから外れてしまうことがないようにカメラをパンまたはチルトする。

#### 【0040】

ステップS8において、制御部27は、ユーザからの指示に基づいて監視処理を終了するか否かを判定し、ユーザから終了が指示されていない場合には、ステップS1に戻り、それ以降の処理を繰り返し実行する。ユーザから監視処理の終了が指示された場合、ステップS8において終了すると判定され、制御部27は監視処理を終了する。

#### 【0041】

図3A乃至Cは、このとき画像ディスプレイ23に表示される画像の例を、時系列に示した図である。図3Aは、撮像部21により、追尾対象となるオブジェクト51が撮像された画像の例であり、この例では、図中左方向に走って移動する人が、オブジェクト51として撮像されている。図3Bでは、オブジェクト51が、図3Aの位置から図中左に移動しており、図3Cでは、オブジェクト51が、図3Bの位置からさらに左に移動している。

#### 【0042】

追尾対象検出部24は、図2のステップS1で、オブジェクト51を検出し、このオブジェクト51（人）の目を、追尾点51Aとしてオブジェクト追尾部26に出力する。ステップS2ではオブジェクト追尾部26により追尾処理が行われ、エリア設定部25により、ステップS3で追尾対象のオブジェクト51（追尾点51A）の周辺の所定領域が補

正対象エリア52として設定される。

【0043】

上述したように、オブジェクト追尾部26は、追尾点51Aに基づいて、オブジェクト51を追尾するので、オブジェクト51が移動すると、追尾点51Aも移動し、その追尾結果（位置）がエリア設定部25に出力される。このため、図3A乃至図3Cに示されるように、オブジェクト51が図中左に移動していくと、補正対象エリア52も図中左に移動していく。

【0044】

移動するオブジェクト51（追尾点51A）に対応する補正対象エリア52の設定は、例えば、次のようにして行われる。図4は、補正対象エリアとして、追尾点の周辺に所定の大きさの四角形の領域が設定される例を示す。同図において、最初に補正対象エリア71Aが設定されたものとする。最初の補正対象エリア71Aとしては、例えば、追尾点51Aを基準として、それを中心とする一定の範囲が設定される。勿論、ユーザにより指定された場合には、その指定された範囲が補正対象エリア71Aとして設定される。このときエリア設定部25は、補正対象エリア71Aの左上の角の座標(X,Y)を、内蔵するメモリに記憶する。オブジェクト51の追尾点51Aが移動すると、オブジェクト追尾部26による追尾が行われ、追尾点51Aの、画面上におけるX軸方向（図中左右の方向）、Y軸方向（図中上下方向）の位置（または移動距離）の情報が、追尾結果としてエリア設定部25に供給される。

【0045】

そして、上述した左上の角の座標に基づいて、補正対象エリアが設定される。例えば、追尾点51Aが画面上で、X軸方向にx、Y軸方向にyだけそれぞれ移動すると、エリア設定部25は、補正対象エリア71Aの左上の座標(X,Y)に、xとyを加算し、座標(X+x, Y+y)を計算し、その座標を新たな補正対象エリア71Bの左上の角の座標として記憶するとともに、補正対象エリア71Bを設定する。追尾点51AがX軸方向にa、Y軸方向にbだけさらに移動すると、エリア設定部25は、補正対象エリア71Aの左上の座標(X+x, Y+y)に、aとbを加算し、座標(X+x+a, Y+y+b)を計算し、その座標を新たな補正対象エリア71Cの左上の角の座標として記憶するとともに、補正対象エリア71Cを設定する。

【0046】

このように、オブジェクト（追尾点）の移動に伴って、補正対象エリアも移動していく。

【0047】

また、上述したように、補正対象エリア52内の画像については、画像補正部22による画像補正処理（図2のステップS4）が行われ、画像のぼけが除去されて、画像ディスプレイ23に表示される。従って、図3A乃至Cに示される画像は、補正対象エリア52内が鮮明に表示され、補正対象エリア52の外の領域（背景）である背景53の画像は、補正対象エリア52内の画像と比較すると、相対的に不鮮明に表示される。

【0048】

このようにすることで、画像ディスプレイ23に表示される画像の中で、補正対象エリア52内のオブジェクト51が、常に鮮明に表示されるので、画像ディスプレイ23を観察するユーザは、自然にオブジェクト51を注目するようになり、その結果、例えば不審者の侵入、物体の移動などをより迅速に発見することができる。また、オブジェクト51は、鮮明に表示されるので、移動するオブジェクト（例えば、人物）が何（誰）であるのか、ユーザに、正確に認識させることができる。

【0049】

次に、図1のオブジェクト追尾部26の詳細な構成例と、その動作について説明する。図5はオブジェクト追尾部26の機能的構成例を示すブロック図である。この例では、オブジェクト追尾部26は、テンプレートマッチング部51、動き推定部52、シーンチェンジ検出部53、背景動き推定部54、領域推定関連処理部55、乗り換え候補保持部5

6、追尾点決定部57、テンプレート保持部58、および制御部59により構成されている。

#### 【0050】

テンプレートマッチング部51は、入力画像と、テンプレート保持部58に保持されているテンプレート画像のマッチング処理を行う。動き推定部52は、入力画像の動きを推定し、推定の結果得られた動きベクトルと、その動きベクトルの確度を、シーンチェンジ検出部53、背景動き推定部54、領域推定関連処理部55、および追尾点決定部57に出力する。シーンチェンジ検出部53は、動き推定部52より供給された確度に基づいて、シーンチェンジを検出する。

#### 【0051】

背景動き推定部54は、動き推定部52より供給された動きベクトルと確度に基づいて背景動きを推定する処理を実行し、推定結果を領域推定関連処理部55に供給する。領域推定関連処理部55は、動き推定部52より供給された動きベクトルと確度、背景動き推定部54より供給された背景動き、並びに追尾点決定部57より供給された追尾点情報に基づいて、領域推定処理を行う。また、領域推定関連処理部55は、入力された情報に基づいて乗り換え候補を生成し、乗り換え候補保持部56へ供給し、保持させる。さらに、領域推定関連処理部55は、入力画像に基づいてテンプレートを作成し、テンプレート保持部58に供給し、保持させる。

#### 【0052】

追尾点決定部57は、動き推定部52より供給された動きベクトルと確度、並びに乗り換え候補保持部56より供給された乗り換え候補に基づいて、追尾点を決定し、決定された追尾点に関する情報を領域推定関連処理部55に出力する。

#### 【0053】

制御部59は、追尾対象検出部から24から出力された追尾点の情報に基づいて、テンプレートマッチング部51乃至テンプレート保持部58の各部を制御して、検出された追尾対象を追尾させるとともに、画像ディスプレイ23に表示される画面の中に、追尾点が表示されるように（追尾点が画面の外にでないように）、カメラ駆動部27に制御信号を出力し、カメラ（撮像部21）の駆動を制御する。また、制御部59は、追尾点の画面上での位置の情報などの追尾結果を、エリア設定部25、制御部27などに出力する。

#### 【0054】

次に、オブジェクト追尾部26の動作について説明する。図6は、図2のステップS2において、オブジェクト追尾部26が実行する追尾処理の詳細を説明するフローチャートである。

#### 【0055】

図7に示されるように、オブジェクト追尾部26は、基本的に通常処理と例外処理を実行する。すなわち、ステップS51で通常処理が行われる。この通常処理の詳細は、図10を参照して後述するが、この処理により追尾対象検出部24により指定された追尾点を追尾する処理が実行される。ステップS51の通常処理において追尾点の乗り換えができなくなったとき、ステップS52において、例外処理が実行される。この例外処理の詳細は、図35のフローチャートを参照して後述するが、この例外処理により、追尾点が画像から見えなくなったとき、テンプレートマッチングにより通常処理への復帰処理が実行されることができなくなった（通常処理へ復帰することができなくなった）と判定された場合には処理が終了されるが、テンプレートによる復帰処理の結果、通常処理への復帰が可能と判定された場合には、処理は再びステップS51に戻る。このようにして、ステップS51の通常処理とステップS52の例外処理が、各フレーム毎に順次繰り返し実行される。

#### 【0056】

本発明においては、この通常処理と例外処理により、図7乃至図9に示されるように、追尾対象が回転したり、オクルージョンが発生したり、シーンチェンジが発生する等、追尾点が一時的に見えなくなった場合においても、追尾が可能となる。

## 【0057】

すなわち、例えば、図7に示されるように、フレーム $n-1$ には追尾対象（オブジェクト）としての人の顔104が表示されており、この人の顔104は、右目102と左目103を有している。ユーザが、このうちの、例えば右目102（正確には、その中の1つの画素）を追尾点101として指定したとする。図7の例においては、次のフレーム $n$ において、人が図中左方向に移動しており、さらに次のフレーム $n+1$ においては、人の顔104が時計方向に回転している。その結果、今まで見えていた右目102が表示されなくなり、いままでの方法では、追尾ができなくなる。そこで、上述したステップS51の通常処理においては、右目102と同一の対象物としての顔104上の左目103が選択され、追尾点が左目103に乗り換えられる（設定される）。これにより追尾が可能となる。

## 【0058】

図8の表示例では、フレーム $n-1$ において、顔104の図中左側からボール121が移動してきて、次のフレーム $n$ においては、ボール121がちょうど顔104を覆う状態となっている。この状態において、追尾点101として指定されていた右目102を含む顔104が表示されていない。このようなオクルージョンが起きると、対象物としての顔104が表示されていないので、追尾点101に代えて追尾する乗り換え点もなくなり、以後、追尾点を追尾することが困難になる。しかし、本発明においては、追尾点101としての右目102をフレーム $n-1$ （実際には時間的にもっと前のフレーム）の画像がテンプレートとして予め保存されており、ボール121がさらに右側に移動し、フレーム $n+1$ において、追尾点101として指定された右目102が再び現れると、上述したステップS52の例外処理により、追尾点101としての右目102が再び表示されたことが確認され、右目102が再び追尾点101として追尾されることになる。

## 【0059】

図9の例では、フレーム $n-1$ においては、顔104が表示されているが、次のフレーム $n$ においては、自動車111が人の顔を含む全体を覆い隠している。すなわち、この場合、シーンチェンジが起きたことになる。本発明では、このようにシーンチェンジが起きて追尾点101が画像から存在しなくなっても、自動車111が移動して、フレーム $n+1$ において再び右目102が表示されると、ステップS52の例外処理で、追尾点101としての右目102が再び出現したことがテンプレートに基づいて確認され、この右目102を再び追尾点101として追尾することが可能となる。

## 【0060】

次に、図10のフローチャートを参照して、図6のステップS51の通常処理の詳細について説明する。ステップS121において、追尾点決定部57により通常処理の初期化処理が実行される。その詳細は、図11のフローチャートを参照して後述するが、この処理によりユーザから追尾するように指定された追尾点を基準とする領域推定範囲が指定される。この領域推定範囲は、ユーザにより指定された追尾点と同一の対象物（例えば、追尾点が人の目である場合、目と同様の動きをする剛体としての人の顔、または人の体など）に属する点の範囲を推定する際に参照する範囲である。乗り換え点は、この領域推定範囲の中の点から選択される。

## 【0061】

次に、ステップS122において、制御部59は、次のフレームの画像の入力を待機するように各部を制御する。ステップS123において、動き推定部52は、追尾点の動きを推定する。すなわち、ユーザにより指定された追尾点を含むフレーム（前フレーム）より時間的に後のフレーム（後フレーム）をステップS122の処理で取り込むことで、結局連続する2フレームの画像が得られたことになるので、ステップS123において、前フレームの追尾点に対応する後フレームの追尾点の位置を推定することで、追尾点の動きが推定される。

## 【0062】

なお、時間的に前とは、処理の順番（入力 of 順番）をいう。通常、撮像の順番に各フレ

ームの画像が入力されるので、その場合、より時間的に前に撮像されたフレームが前フレームとなるが、時間的に後に撮像されたフレームが先に処理（入力）される場合には、時間的に後に撮像されたフレームが前フレームとなる。

#### 【0063】

ステップS124において、動き推定部52は、ステップS123の処理の結果、追尾点が推定可能であったか否かを判定する。追尾点が推定可能であったか否かは、例えば、動き推定部52が生成、出力する動きベクトル（後述）の確度の値を、予め設定されている閾値と比較することで判定される。具体的には、動きベクトルの確度が閾値以上であれば推定が可能であり、閾値より小さければ推定が不可能であると判定される。すなわち、ここにおける可能性は比較的厳格に判定され、実際には推定が不可能ではなくても確度が低い場合には、不可能と判定される。これにより、より確実な追尾処理が可能となる。

#### 【0064】

なお、ステップS124では、追尾点での動き推定結果と追尾点の近傍の点での動き推定結果が、多数を占める動きと一致する場合には推定可能、一致しない場合には推定不可能と判定するようにすることも可能である。

#### 【0065】

追尾点の動きが推定可能であると判定された場合（追尾点が同一対象物上の対応する点上に正しく設定されている確率（右目102が追尾点101として指定された場合、右目102が正しく追尾されている確率）が比較的高い場合）、ステップS125に進み、追尾点決定部57は、ステップS123の処理で得られた推定動き（動きベクトル）の分だけ追尾点をシフトする。すなわち、これにより、前フレームの追尾点の追尾後の後フレームにおける追尾の位置が決定されることになる。

#### 【0066】

ステップS25の処理の後、ステップS126において、領域推定関連処理が実行される。この領域推定関連処理の詳細は、図14を参照して後述するが、この処理により、ステップS121の通常処理の初期化処理で指定された領域推定範囲が更新される。さらに、対象物体が回転するなどして、追尾点が表示されない状態になった場合に、追尾点を乗り換えるべき点としての乗り換え点としての候補（乗り換え候補）が、この状態（まだ追尾が可能な状態）において、予め抽出（作成）される。また、乗り換え候補への乗り換えもできなくなった場合、追尾は一旦中断されるが、再び追尾が可能になった（追尾点が再び出現した）ことを確認するために、テンプレートが予め作成される。

#### 【0067】

ステップS126の領域推定関連処理が終了した後、処理は再びステップS122に戻り、それ以降の処理が繰り返し実行される。

#### 【0068】

すなわち、ユーザから指定された追尾点の動きが推定可能である限り、ステップS122乃至ステップS126の処理がフレーム毎に繰り返し実行され、追尾が行われることになる。

#### 【0069】

これに対して、ステップS124において、追尾点の動きが推定可能ではない（不可能である）と判定された場合、すなわち、上述したように、例えば動きベクトルの確度が閾値以下であるような場合、処理はステップS127に進む。ステップS127において、追尾点決定部57は、ステップS126の領域推定関連処理で生成された乗り換え候補が乗り換え候補保持部116に保持されているので、その中から、元の追尾点に最も近い乗り換え候補を1つ選択する。追尾点決定部57は、ステップS128で乗り換え候補が選択できたか否かを判定し、乗り換え候補が選択できた場合には、ステップS129に進み、追尾点をステップS127の処理で選択した乗り換え候補に乗り換える（変更する）。すなわち、乗り換え候補の点が新たな追尾点として設定される。その後、処理はステップS123に戻り、乗り換え候補の中から選ばれた追尾点の動きを推定する処理が実行される。



## 【0070】

ステップS124において新たに設定された追尾点の動きが推定可能であるか否かが再び判定され、推定可能であれば、ステップS125において追尾点を推定動き分だけシフトする処理が行われ、ステップS126において、領域推定関連処理が実行される。その後、処理は再びステップS122に戻り、それ以降の処理が繰り返し実行される。

## 【0071】

ステップS124において、新たに設定された追尾点も推定不可能であると判定された場合には、再びステップS127に戻り、乗り換え候補の中から、元の追尾点に次に最も近い乗り換え候補が選択され、ステップS129において、その乗り換え候補が新たな追尾点とされる。その新たな追尾点について、再びステップS123以降の処理が繰り返される。

## 【0072】

用意されているすべての乗り換え候補を新たな追尾点としても、追尾点の動きを推定することができなかった場合には、ステップS128において、乗り換え候補が選択できなかったと判定され、この通常処理は終了される。そして、図6のステップS52の例外処理に処理が進むことになる。

## 【0073】

次に、図11のフローチャートを参照して、図10のステップS121の通常処理の初期化処理の詳細について説明する。

## 【0074】

ステップS141において、制御部59は、今の処理は例外処理からの復帰の処理であるのか否かを判定する。すなわち、ステップS52の例外処理を終了した後、再びステップS51の通常処理に戻ってきたのか否かが判定される。最初のフレームの処理においては、まだステップS52の例外処理は実行されていないので、例外処理からの復帰ではないと判定され、処理はステップS142に進む。ステップS142において、追尾点決定部57は、追尾点を追尾点指示の位置に設定する処理を実行する。すなわち、入力画像の中で、例えば、最も輝度の高い点が追尾点として設定される。なお、追尾点は、ユーザによる指定など、他の方法により設定されるようにしてもよい。追尾点決定部57は、設定した追尾点の情報を領域推定関連処理部55に供給する。

## 【0075】

ステップS143において、領域推定関連処理部55は、ステップS142の処理で設定された追尾点の位置に基づき、領域推定範囲を設定する。この領域推定範囲は、追尾点と同じ剛体上の点を推定する際の参照範囲であり、予め追尾点と同じ剛体部分が領域推定範囲の大部分を占めるように、より具体的には、追尾点と同じ剛体部分に推定領域範囲の位置や大きさが追従するように設定することで、領域推定範囲の中で最も多数を占める動きを示す部分を追尾点と同じ剛体部分であると推定できるようにするためのものである。ステップS143では初期値として、例えば、追尾点を中心とする予め設定された一定の範囲が領域推定範囲とされる。

## 【0076】

その後処理は、図10のステップS122に進むことになる。

## 【0077】

一方、ステップS141において、現在の処理が、ステップS52の例外処理からの復帰の処理であると判定された場合、ステップS144に進み、追尾点決定部57は、後述する図35を参照して後述する処理により、テンプレートにマッチした位置に基づき追尾点と領域推定範囲を設定する。例えば、テンプレート上の追尾点とマッチした現フレーム上の点が追尾点とされ、その点から予め設定されている一定の範囲が領域推定範囲とされる。その後、処理は図10のステップS122に進む。

## 【0078】

以上の処理を図12を参照して説明すると次のようになる。すなわち、図11のステップS142において、例えば、図12に示されるように、フレームn-1の人の目102

が追尾点101として指定されると、ステップS143において、追尾点101を含む所定の領域が領域推定範囲133として指定される。ステップS124において、領域推定範囲133の範囲内のサンプル点が次のフレームにおいて推定可能であるか否かが判定される。図12の例の場合、フレームnの次のフレームn+1においては、領域推定範囲133のうち、左目102を含む図中左側半分の領域134がボール121で隠されているため、フレームnの追尾点101の動きを、次のフレームn+1において推定することができない。そこで、このような場合においては、時間的に前のフレームn-1で乗り換え候補として予め用意されていた領域指定範囲133内（右目102を含む剛体としての顔104内）の点の中から1つの点（例えば、顔104に含まれる左目103（正確には、その中の1つの画素））が選択され、その点がフレームn+1における、追尾点とされる。

#### 【0079】

領域推定関連処理部55は、図10のステップS126における領域推定関連処理を実行するために、図13に示されるような構成を有している。すなわち、領域推定関連処理部55の領域推定部161には、動き推定部52より動きベクトルと確度が入力され、背景動き推定部54より背景動きが入力され、そして追尾点決定部57より追尾点の位置情報が入力される。乗り換え候補抽出部162には、動き推定部52より動きベクトルと確度が供給される他、領域推定部161の出力が供給される。テンプレート作成部163には、入力画像が入力される他、領域推定部161の出力が入力される。

#### 【0080】

領域推定部161は、入力に基づいて、追尾点を含む剛体の領域を推定し、推定結果を乗り換え候補抽出部162とテンプレート作成部163に出力する。乗り換え候補抽出部162は入力に基づき乗り換え候補を抽出し、抽出した乗り換え候補を乗り換え候補保持部56へ供給する。テンプレート作成部163は入力に基づきテンプレートを作成し、作成したテンプレートをテンプレート保持部58へ供給する。

#### 【0081】

図14は、領域推定関連処理部55により実行される領域推定関連処理（図10のステップS126の処理）の詳細を表している。最初にステップS161において、領域推定部161により領域推定処理が実行される。その詳細は、図15のフローチャートを参照して後述するが、この処理により、追尾点が属する対象と同一の対象（追尾点と同期した動きをする剛体）に属すると推定される画像上の領域の点が領域推定範囲の点として抽出される。

#### 【0082】

ステップS162において、乗り換え候補抽出部162により乗り換え候補抽出処理が実行される。その処理の詳細は、図25のフローチャート参照して後述するが、領域推定部161により領域推定範囲として推定された範囲の点から乗り換え候補の点が抽出され、乗り換え候補保持部56に保持される。

#### 【0083】

ステップS163においてテンプレート作成部163によりテンプレート作成処理が実行される。その詳細は、図26のフローチャートを参照して後述するが、この処理によりテンプレートが作成される。

#### 【0084】

次に、図15のフローチャートを参照して、図14のステップS161の領域推定処理の詳細について説明する。

#### 【0085】

最初に、ステップS181において、領域推定部161は、追尾点と同一の対象に属すると推定される点の候補の点としてのサンプル点を決定する。

#### 【0086】

このサンプル点は、例えば図16に示されるように、図中、白い四角形で示されるフレームの全画面における画素のうち、固定された基準点201を基準として、水平方向およ

び垂直方向に、所定の画素数ずつ離れた位置の画素をサンプル点（図中、黒い四角形で表されている）とすることができる。図16の例においては、各フレームの左上の画素が基準点201とされ（図中基準点201は×印で示されている）、水平方向に5個、並びに垂直方向に5個ずつ離れた位置の画素がサンプル点とされる。すなわち、この例の場合、全画面中に分散した位置の画素がサンプル点とされる。また、この例の場合、基準点は、各フレーム  $n$ 、 $n+1$  において固定された同一の位置の点とされる。

#### 【0087】

基準点201は、例えば図17に示されるように、各フレーム  $n$ 、 $n+1$  毎に異なる位置の点となるように、動的に変化させることもできる。

#### 【0088】

図16と図17の例においては、サンプル点の間隔が各フレームにおいて固定された値とされているが、例えば図18に示されるように、フレーム毎にサンプル点の間隔を可変とすることもできる。図18の例においては、フレーム  $n$  においては、サンプル点の間隔は5画素とされているのに対し、フレーム  $n+1$  においては8画素とされている。このときの間隔の基準としては、追尾点と同一の対象に属すると推定される領域の面積を用いることができる。具体的には、領域推定範囲の面積が狭くなれば間隔も短くなる。

#### 【0089】

あるいはまた、図19に示されるように、1つのフレーム内においてサンプル点の間隔を可変とすることもできる。このときの間隔の基準としては、追尾点からの距離を用いることができる。すなわち、追尾点に近いサンプル点ほど間隔が小さく、追尾点から遠くなるほど間隔が大きくなる。

#### 【0090】

以上のようにしてサンプル点が決定されると、次にステップS182において、領域推定部161は、領域推定範囲（図11のステップS143、S144の処理、または、後述する図20のステップS206、S208の処理で決定されている）内のサンプル点の動きを推定する処理を実行する。すなわち、領域推定部161は、動き推定部52より供給された動きベクトルに基づいて、領域推定範囲内のサンプル点に対応する次のフレームの対応する点を抽出する。

#### 【0091】

ステップS183において、領域推定部161は、ステップS182の処理で推定したサンプル点のうち、確度が予め設定されている閾値より低い動きベクトルに基づく点を対象外とする処理を実行する。この処理に必要な動きベクトルの確度は、動き推定部52より供給される。これにより、領域推定範囲内のサンプル点のうち、確度が高い動きベクトルに基づいて推定された点だけが抽出される。

#### 【0092】

ステップS184において、領域推定部161は、領域推定範囲内の動き推定結果での全画面動きを抽出する。全画面動きとは、同一の動きに対応する領域を考え、その面積が最大となる動きのことを意味する。具体的には、各サンプル点の動きに、そのサンプル点におけるサンプル点間隔に比例する重みを付けて動きのヒストグラムを生成し、この重み付け頻度が最大となる1つの動き（1つの動きベクトル）が全画面動きとして抽出される。なお、ヒストグラムを生成する場合、例えば、動きの代表値を画素精度で準備し、画素精度で1個となる値を持つ動きについてもヒストグラムへの加算を行うようにすることもできる。

#### 【0093】

ステップS185において、領域推定部161は、全画面動きを持つ領域推定範囲内のサンプル点を領域推定の結果として抽出する。この場合における全画面動きを持つサンプル点としては、全画面動きと同一の動きを持つサンプル点はもちろんのこと、全画面動きとの動きの差が予め設定されている所定の閾値以下である場合には、そのサンプル点もここにおける全画面動きを持つサンプル点とすることも可能である。

#### 【0094】

このようにして、ステップS143, S144, S206, S208の処理で決定された領域推定範囲内のサンプル点のうち、全画面動きを有するサンプル点が、追尾点と同一対象に属すると推定される点として最終的に抽出（生成）される。

【0095】

次に、ステップS186において、領域推定部161は、領域推定範囲の更新処理を実行する。その後、処理は、図10のステップS122に進む。

【0096】

図20は、図15のステップS186の領域推定範囲の更新処理の詳細を表している。ステップS201において、領域推定部161は、領域の重心を算出する。この領域とは、図15のステップS185の処理で抽出されたサンプル点で構成される領域（追尾点と同一対象に属すると推定される点で構成される領域）を意味する。すなわち、この領域には1つの動きベクトル（全画面動き）が対応している。例えば、図21Aに示されるように、図中白い四角形で示されるサンプル点のうち、領域推定範囲221内のサンプル点の中から、図15のステップS185の処理で全画面動きを持つサンプル点として、図21Aにおいて黒い四角形で示されるサンプル点が抽出され、そのサンプル点で構成される領域が、領域222として抽出（推定）される。そして、領域222の重心224がさらに算出される。具体的には、各サンプル点にサンプル点間隔の重みを付けたサンプル点重心が領域の重心として求められる。この処理は、現フレームにおける領域の位置を求めるという意味を有する。

【0097】

次にステップS202において、領域推定部161は、領域の重心を全画面動きによりシフトする処理を実行する。この処理は、領域推定範囲221を領域の位置の動きに追従させ、次フレームにおける推定位置に移動させるという意味を有する。図21Bに示されるように、現フレームにおける追尾点223が、その動きベクトル238に基づいて次フレームにおいて追尾点233として出現する場合、全画面動きベクトル230が、追尾点の動きベクトル238にはほぼ対応しているので、現フレームにおける重心224を動きベクトル230（全画面動き）に基づいてシフトすることで、追尾点233と同一のフレーム（次フレーム）上の点234が求められる。この点234を中心として領域推定範囲231を設定すれば、領域推定範囲221を領域222の位置の動きに追従させて、次のフレームにおける推定位置に移動させることになる。

【0098】

ステップS203において、領域推定部161は、領域推定結果に基づき、次の領域推定範囲の大きさを決定する。具体的には、領域と推定された全てのサンプル点に関するサンプル点の間隔（図21Aにおける領域222の中の黒い四角形で示される点の間隔）の2乗和を領域222の面積と見なし、この面積よりも少し大きめの大きさとなるように、次フレームにおける領域推定範囲231の大きさが決定される。すなわち、領域推定範囲231の大きさは、領域222の中のサンプル点の数が多ければ広くなり、少なければ狭くなる。このようにすることで、領域222の拡大縮小に追従することができるばかりでなく、領域推定範囲221内の全画面領域が追尾対象の周辺領域となるのを防ぐことができる。

【0099】

図15のステップS184で抽出された全画面動きが、背景動きと一致する場合には、動きにより背景と追尾対象を区別することができない。そこで、背景動き推定部54は背景動き推定処理を常に行っており、ステップS204において、領域推定部161は、背景動き推定部54より供給される背景動きと、図15のステップS184の処理で抽出された全画面動きとが一致するか否かを判定する。全画面動きと背景動きが一致する場合には、ステップS205において、領域推定部161は、次の領域推定範囲の大きさを、今の領域推定範囲の大きさが最大となるように制限する。これにより、背景が追尾対象として誤認識され、領域推定範囲の大きさが拡大してしまうようなことが抑制される。

【0100】

ステップ S 2 0 4 において、全画面動きと背景動きが一致しないと判定された場合には、ステップ S 2 0 5 の処理は必要がないのでスキップされる。

【0101】

次に、ステップ S 2 0 6 において、領域推定部 1 6 1 は、シフト後の領域重心を中心として次の領域推定範囲の大きさを決定する。これにより、領域推定範囲が、その重心が既に求めたシフト後の領域重心と一致し、かつ、その大きさが領域の広さに比例するように決定される。

【0102】

図 2 1 B の例では、領域推定範囲 2 3 1 が、動きベクトル（全画面動き） 2 3 0 に基づくシフト後の重心 2 3 4 を中心として、領域 2 2 2 の面積に応じた広さに決定されている。

【0103】

領域推定範囲 2 3 1 内での全画面動きを有する領域が追尾対象（例えば、図 1 2 の顔 1 0 4）の領域であることを担保する（確実にする）必要がある。そこで、ステップ S 2 0 7 において、領域推定部 1 6 1 は、追尾点が次の領域推定範囲に含まれるか否かを判定し、含まれていない場合には、ステップ S 2 0 8 において、追尾点を含むように次の領域推定範囲をシフトする処理を実行する。追尾点が次の領域推定範囲に含まれている場合には、ステップ S 2 0 8 の処理は必要がないのでスキップされる。

【0104】

この場合における具体的なシフトの方法としては、移動距離が最小となるようにする方法、シフト前の領域推定範囲の重心から追尾点に向かうベクトルに沿って追尾点が含まれるようになる最小距離だけ移動する方法などが考えられる。

【0105】

なお、追尾のロバスト性を重視するために、領域に追尾点を含むようにするためのシフトを行わない方法も考えられる。

【0106】

図 2 1 C の例においては、領域推定範囲 2 3 1 が追尾点 2 3 3 を含んでいないので、領域推定範囲 2 4 1 として示される位置（追尾点 2 3 3 をその左上に含む位置）に領域推定範囲 2 4 1 がシフトされる。

【0107】

図 2 1 A 乃至図 2 1 C は、ステップ S 2 0 8 のシフト処理が必要な場合を示しているが、図 2 2 A 乃至図 2 2 C は、ステップ S 2 0 8 のシフト処理が必要でない場合（ステップ S 2 0 7 において追尾点が次の領域推定範囲に含まれると判定された場合）の例を表している。

【0108】

図 2 2 A 乃至図 2 2 C に示されるように、領域推定範囲 2 2 1 内のすべてのサンプル点が領域の点である場合には、図 2 0 のステップ S 2 0 8 のシフト処理が必要なくなることになる。

【0109】

なお、図 2 1 A 乃至図 2 1 C と図 2 2 A 乃至図 2 2 C は、領域推定範囲が矩形である例を示したが、領域推定範囲は、他の形状（例えば、円形）としてもよい。

【0110】

以上のようにして、図 2 0（図 1 5 のステップ S 1 8 6）の領域推定範囲の更新処理により、次フレームのための領域推定範囲の位置と大きさが追尾点を含むように決定される。

【0111】

図 2 0 の領域推定範囲の更新処理においては、領域推定範囲を矩形（または円形）の固定形状としたが、可変形状とすることも可能である。この場合における図 1 5 のステップ S 1 8 6 における領域推定範囲の更新処理の例について、図 2 3 を参照して説明する。

【0112】

ステップS231において、領域推定部161は、図15のステップS184の処理で抽出された全画面動きと背景動き推定部164により推定された背景動きとが一致するかどうかを判定する。両者の動きが一致しない場合には、ステップS233に進み、領域推定部161は、領域（全画面動きと一致する画素で構成される領域）と推定されたすべての点につき、それぞれに対応する小領域を決定する（1個の点に対して1個の小領域を決定する）。図24Aと図24Bの例においては、領域推定範囲261のうち、図中黒い四角形で示される領域の点に対応する小領域271, 272が決定される。図中小領域271は、4つの点に対応する小領域が重なりあった例を示している。小領域の大きさは、例えば、サンプル点の間隔に比例するように決定してもよい。

#### 【0113】

次に、ステップS234において、領域推定部161は、ステップS233の処理で決定した各小領域の和の領域を暫定領域推定範囲とする。図24Cの例においては、小領域271と小領域272の和の領域281が暫定領域推定範囲とされる。小領域の和をとった結果、不連続な複数の領域が形成される場合には、その中の最大面積を持つ領域のみを暫定領域推定範囲とすることもできる。

#### 【0114】

ステップS231において、全画面動きと背景動きとが一致すると判定された場合には、ステップS232において、領域推定部161は、現在の領域推定範囲を暫定領域推定範囲とする。現在の領域推定範囲を暫定領域推定範囲とするのは、背景動き推定の結果と全画面動きが一致する場合には、動きにより背景と追尾対象を区別することができないので、現在の領域推定範囲を変更しないようにするためである。

#### 【0115】

ステップS234またはステップS232の処理の後、ステップS235において領域推定部161は、ステップS234またはステップS232で決定された暫定領域推定範囲の全画面動きによるシフトで、次の領域推定範囲を決定する。図24Dの例においては、暫定領域推定範囲281が全画面動きによる動きベクトル283に基づいてシフトされ、暫定領域推定範囲282とされている。

#### 【0116】

ステップS236において、領域推定部161は、追尾点がステップS235の処理で決定された次の領域推定範囲に含まれるかどうかを判定し、含まれない場合には、ステップS237に進み、追尾点を含むように次の領域推定範囲をシフトする。図24Cと図24Dの例においては、領域推定範囲282が追尾点284を含んでいないので、追尾点284を左上に含むようにシフトされ、領域推定範囲291とされている。

#### 【0117】

ステップS236において、追尾点が次の領域推定範囲に含まれると判定された場合には、ステップS237のシフト処理は必要ないのでスキップされる。

#### 【0118】

次に図14のステップS162における乗り換え候補抽出処理について、図25のフローチャートを参照して説明する。

#### 【0119】

ステップS261において、乗り換え候補抽出部162は、全画面動きの領域と推定されたすべての点につき、それぞれに対応する推定動きでの点のシフト結果を乗り換え候補として保持する。すなわち、領域推定結果として得られた点をそのまま用いるのではなく、それらを次のフレームでの使用のために、それぞれの動き推定結果に基づきシフトされた結果を抽出する処理が行われ、その抽出された乗り換え候補が、乗り換え候補保持部56に供給され、保持される。

#### 【0120】

この処理を図12を参照して説明すると、次のようになる。すなわち、図12の例において、フレーム $n-1$ 、 $n$ では追尾点101が存在するが、フレーム $n+1$ においては、図中左側から飛んできたボール121により隠されてしまい、追尾点101が存在しない

。そこでフレーム  $n+1$  において、追尾点を追尾対象としての顔 104 上の他の点（例えば、左目 103（実際には右目 102 にもっと近接した点））に乗り換える必要が生じる。そこで、乗り換えが実際に必要になる前のフレームで、乗り換え候補を予め用意しておくのである。

#### 【0121】

具体的には、図 12 の例の場合、フレーム  $n$  からフレーム  $n+1$  への領域推定範囲 133 内での動き推定結果は、領域推定範囲 133 において乗り換えが必要なことから、正しく推定できない確率が高いことが予想される。すなわち、図 12 の例では、乗り換えが追尾点と、それと同一の対象物の一部が隠れることに起因して起きる。その結果、フレーム  $n$  での領域推定範囲 133 のうち、フレーム  $n+1$  で対象が隠れる部分（図 12 において影を付した部分）134 については、動きが正しく推定されず、動きの確度が低いことが推定されるか、または確度が低くないと推定され、かつ、動き推定結果としては意味のないものが得られることになる。

#### 【0122】

このような場合には、領域推定の際に用いることが可能な動き推定結果が減少する、あるいは誤った動き推定結果が混入するなどの理由で、領域推定が誤る可能性が高まる。一方、このような可能性は、一般的に、より時間的に前のフレーム  $n-1$  からフレーム  $n$  の間での領域推定においては、フレーム  $n$  からフレーム  $n+1$  での間での推定に比較して低くなることが予想される。

#### 【0123】

そこで、リスク低減のため、領域推定結果をそのまま用いるのではなく、前のフレーム  $n-1$ （あるいは、時間的にもっと前のフレーム）で求めた領域推定結果を、その次のフレームでの移動先の乗り換え候補として用いるのが性能向上の上で望ましい。

#### 【0124】

図 26 は、図 14 のステップ S163 におけるテンプレート作成処理の詳細を表している。ステップ S281 においてテンプレート作成部 163 は、領域（全画面動きの領域）と推定されたすべての点につき、それぞれに対応する小領域を決定する。図 27 の例においては、領域の点 321 に対応して小領域 322 が決定されている。

#### 【0125】

ステップ S282 において、テンプレート作成部 163 は、ステップ S281 の処理で決定された小領域の和の領域をテンプレート範囲に設定する。図 27 の例においては、小領域 322 の和の領域がテンプレート範囲 331 とされている。

#### 【0126】

次にステップ S283 において、テンプレート作成部 163 は、ステップ S282 において設定したテンプレート範囲の情報と画像情報からテンプレートを作成し、テンプレート保持部 58 に供給し、保持させる。具体的には、テンプレート範囲 331 内の画素データがテンプレートとされる。

#### 【0127】

図 28 は、領域の点 321 に対応する小領域 341 が、図 27 における小領域 322 に較べてより大きな面積とされている。その結果、小領域 341 の和の領域のテンプレート範囲 351 も、図 27 のテンプレート範囲 331 に較べてより広がっている。

#### 【0128】

小領域の大きさは、サンプル点の間隔に比例させることが考えられるが、その際の比例定数は、面積がサンプル点間隔の自乗になるように決めることもできるし、それより大きくまたは小さく決めることも可能である。

#### 【0129】

なお、領域推定結果を用いず、例えば追尾点を中心とする固定の大きさや形状の範囲をテンプレート範囲として用いることも可能である。

#### 【0130】

図 29 は、テンプレートと領域推定範囲の位置関係を表している。テンプレート範囲 4

03には、追尾点405が含まれている。テンプレート範囲403に外接する外接矩形401の図中左上の点がテンプレート基準点404とされている。テンプレート基準点404から追尾点405に向かうベクトル406、並びにテンプレート基準点404から領域推定範囲402の図中左上の基準点408に向かうベクトル407が、テンプレート範囲403の情報とされる。テンプレートは、テンプレート範囲403に含まれる画素で構成される。ベクトル406、407は、テンプレートと同じ画像が検出された際の通常処理への復帰に用いられる。

#### 【0131】

以上の処理においては、乗り換え候補の場合と異なり、範囲、画素ともに、現フレームに対応するものをテンプレートとする例を説明したが、乗り換え候補の場合と同様に、次フレームでの移動先をテンプレートとして用いることも可能である。

#### 【0132】

以上のようにして、追尾点を含む画素データからなるテンプレートが乗り換え候補と同様に、通常処理中に、予め作成される。

#### 【0133】

図10のステップS126における領域推定関連処理は、領域推定関連処理部55を、例えば図30に示されるように構成することで処理することも可能である。

#### 【0134】

この場合においても、領域推定関連処理部55は、図13における場合と同様に、領域推定部161、乗り換え候補抽出部162、およびテンプレート作成部163により構成されるが、この実施の形態の場合、領域推定部161には、追尾点決定部57より追尾点の情報と入力画像が入力される。乗り換え候補抽出部162には、領域推定部161の出力のみが供給されている。テンプレート作成部163には、領域推定部161の出力と入力画像とが供給されている。

#### 【0135】

この場合においても、基本的な処理は、図14に示される場合と同様に、ステップS161において、領域推定処理が行われ、ステップS162において、乗り換え候補抽出処理が行われ、ステップS163において、テンプレート作成処理が行われる。このうちのステップS163のテンプレート作成処理は、図26に示した場合と同様であるので、ステップS161の領域推定処理と、ステップS162の乗り換え候補抽出処理についてのみに以下に説明する。

#### 【0136】

最初に、図31のフローチャートを参照して、ステップS161における領域推定処理の詳細について説明する。ステップS301において、図30の領域推定部161は、追尾点と同一対象に属する画像上の領域を推定するためにサンプル点を決定する。この処理は、図15のステップS181の処理と同様の処理である。

#### 【0137】

ただし、このステップS301の処理において対象とされるフレームは、追尾点を求め終わったフレーム（追尾後の追尾点を含むフレーム）であり、この点、図15のステップS181においてサンプル点を求めるフレームが前フレームであるのと異なる。

#### 【0138】

次に、このステップS302において、領域推定部161は、次フレーム（ステップS301でサンプル点を決定したフレーム）の画像に空間方向のローパスフィルタを施す処理を実行する。すなわちローパスフィルタを施すことにより、高周波成分が除去され、画像が平滑化される。これにより、次のステップS303における同色領域の成長処理が容易になる。

#### 【0139】

次に、ステップS303において、領域推定部161は、追尾点を出発点として、画素値の差分が閾値THing未満であるという条件で、追尾点の同色領域を成長させ、同色領域に含まれるサンプル点を領域の推定結果とする処理を実行する。領域の推定結果としては



、成長させた結果の同色領域に含まれるサンプル点が利用される。

【0140】

具体的には、例えば図32Aに示されるように、追尾点421に隣接する8個の方向の画素の画素値が読み取られる。すなわち、上方向、右上方向、右方向、右下方向、下方向、左下方向、左方向、および左上方向の8つの方向に隣接する画素の画素値が読み取られる。読み取られた画素値と追尾点421の画素値との差分が演算される。そして、演算された差分値が閾値THing以上であるか否かが判定される。図32Aの場合、矢印を付して示される方向の画素値、すなわち上方向、右上方向、下方向、左方向、および左上方向の画素値と追尾点321の差分が閾値THing未満であり、図中矢印を付さず示されている方向、すなわち右方向、右下方向、および左下方向の画素値と追尾点421の差分が閾値THing以上であったとされる。

【0141】

この場合、図32Bに示されるように、差分が閾値THing未満である画素（図32Aにおいて、追尾点421に対して矢印で示される方向の画素）が追尾点421と同色領域の画素422として登録される。同様の処理が同色領域に登録された各画素422において行われる。図32Bに示される例では、図中左上の白い円で示される画素422とそれに隣接する画素（既に同色領域であるとの判定が行われた画素を除く）の画素値の差分が演算され、その差分が閾値THing以上であるか否かが判定される。図32Bの例においては、右方向、右下方向、および下方向の画素は、既に同色領域の判定処理が終了している方向なので、上方向、右上方向、左下方向、左方向、および左上方向においての差分が演算される。そして、この例では、上方向、右上方向、および左上方向の3つの方向の差分が閾値THing未満とされ、図32Cに示されるように、その方向の画素が追尾点421と同色領域の画素として登録される。

【0142】

以上のような処理が順次繰り返されることで、図33に示されるように、サンプル点のうち、同色領域431に含まれる点が追尾点421と同一対象物上の点として推定される。

【0143】

図31に示される領域推定処理（図14のステップS161）に続いて、図30の乗り換え候補抽出部162において実行される図14のステップS162の乗り換え候補抽出処理は、図34のフローチャートに示されるようになる。

【0144】

すなわち、ステップS331において、乗り換え候補抽出部162は、領域（同色領域）と推定されたすべての点をそのまま乗り換え候補とし、それを乗り換え候補保持部56に供給し、保持させる。

【0145】

図30の領域推定関連処理部55において、図31の領域推定処理（図14のステップS161）、図34の乗り換え候補抽出処理（図14のステップS162）に引き続き、図30のテンプレート作成部163で実行される図14のステップS163のテンプレート作成処理は、図26に示される場合と同様であるので、その説明は省略する。

【0146】

ただし、この場合においては、追尾点の同色領域をそのままテンプレートの範囲とすることも可能である。

【0147】

以上に説明した図6のステップS51の通常処理に続いて行われるステップS52の例外処理の詳細について、図35のフローチャートを参照して説明する。この処理は、上述したように、図10のステップS124において追尾点の動きを推定することが不可能と判定され、さらにステップS128において追尾点を乗り換える乗り換え候補が選択できなかったと判定された場合に実行されることになる。

【0148】

ステップS401において、制御部59は、例外処理の初期化処理を実行する。この処理の詳細は図36のフローチャートに示されている。

#### 【0149】

ステップS321において、制御部59は、追尾点の追尾ができなくなった際（追尾点の動きを推定することが不可能かつ、追尾点を乗り換える乗り換え候補が選択できなかった際）にシーンチェンジが起きていたか否かを判定する。シーンチェンジ検出部53は、動き推定部52の推定結果に基づいてシーンチェンがあったか否かを常に監視しており、制御部59は、そのシーンチェンジ検出部53の検出結果に基づいて、ステップS421の判定を実行する。シーンチェンジ検出部53の具体的処理については、図47と図48を参照して後述する。

#### 【0150】

シーンチェンジが起きている場合、追尾ができなくなった理由がシーンチェンジが発生したことによるものと推定して、ステップS422において制御部19は、モードをシーンチェンジに設定する。これに対して、ステップS421においてシーンチェンジが発生していないと判定された場合には、制御部59は、ステップS423においてモードをその他のモードに設定する。

#### 【0151】

ステップS422またはステップS423の処理の後、ステップS424においてテンプレートマッチング部51は、時間的に最も古いテンプレートを選択する処理を実行する。具体的には、図37に示されるように、例えばフレームnからフレームn+1に移行するとき、例外処理が実行されるものとする、フレームn-m+1からフレームnに関して生成され、テンプレート保持部58に保持されているm個のフレームのテンプレートの中から、時間的に最も古いテンプレートであるフレームn-m+1に関して生成されたテンプレートが選択される。

#### 【0152】

このように例外処理への移行直前のテンプレート（図37の例の場合フレームnに関して生成されたテンプレート）を用いずに、時間的に少し前のテンプレートを選択するのは、追尾対象のオクルージョンなどで例外処理への移行が発生した場合には、移行の直前には追尾対象が既にかなり隠れており、その時点のテンプレートでは、追尾対象を十分に大きく捉えることができない可能性が高いからである。従って、このように時間的に若干前のフレームにおけるテンプレートを選択することで、確実な追尾が可能となる。

#### 【0153】

次に、ステップS425において、テンプレートマッチング部51は、テンプレート探索範囲を設定する処理を実行する。テンプレート探索範囲は、例えば、例外処理に移行する直前の追尾点の位置がテンプレート探索範囲の中心となるように設定される。

#### 【0154】

すなわち、図38に示されるように、フレームnにおいて被写体の顔104の右目102が追尾点101として指定されている場合において、図中左方向からボール121が飛んできて、フレームn+1において追尾点101を含む顔104が隠れ、フレームn+2において、再び追尾点101が現れる場合を想定する。この場合において、追尾点101（テンプレート範囲411に含まれる）を中心とする領域がテンプレート探索範囲412として設定される。

#### 【0155】

ステップS426において、テンプレートマッチング部51は、例外処理への移行後の経過フレーム数およびシーンチェンジ数を0にリセットする。このフレーム数とシーンチェンジ数は、後述する図35のステップS405における継続判定処理（図39のステップS461、S463、S465、S467）において使用される。

#### 【0156】

以上のようにして、例外処理の初期化処理が終了した後、図35のステップS402において、制御部59は次のフレームを待つ処理を実行する。ステップS403において、

テンプレートマッチング部51は、テンプレート探索範囲内においてテンプレートマッチング処理を行う。ステップS404においてテンプレートマッチング部51は、通常処理への復帰が可能であるか否かを判定する。

**【0157】**

具体的には、テンプレートマッチング処理により、数フレーム前のテンプレート（図38のテンプレート範囲411内の画素）と、テンプレート探索範囲内のマッチング対象の画素の差分の絶対値和が演算される。より詳細には、テンプレート範囲411内の所定のブロックと、テンプレート探索範囲内の所定のブロックにおけるそれぞれの画素の差分の絶対値和が演算される。ブロックの位置がテンプレート範囲411内で順次移動され、各ブロックの差分の絶対値和が加算され、そのテンプレートの位置における値とされる。そして、テンプレートをテンプレート探索範囲内で順次移動させた場合における差分の絶対値和が最も小さくなる位置とその値が検索される。ステップS404において、最小の差分の絶対値和が、予め設定されている所定の閾値と比較される。差分の絶対値和が閾値以下である場合には、追尾点（テンプレートに含まれている）を含む画像が再び出現したことになるので、通常処理への復帰が可能であると判定され、処理は図6のステップS51の通常処理に戻る。

**【0158】**

そして上述したように、図11のステップS141において、例外処理からの復帰であると判定され、ステップS144において、差分絶対値和が最小となる位置をテンプレートのマッチした位置として、このマッチした位置とテンプレートに対応して保持してあったテンプレート位置と追尾点領域推定範囲の位置関係から、追尾点と領域推定範囲の設定が行われる。すなわち、図29を参照して説明したように、追尾点405を基準とするベクトル406、407に基づいて、領域推定範囲402が設定される。

**【0159】**

ただし、図14のステップS161の領域推定処理において、領域推定範囲を用いない手法を用いる場合（例えば、図31に示される領域推定処理が用いられる場合）には、領域推定範囲の設定は行われない。

**【0160】**

図35のステップS404における通常処理への復帰が可能であるか否かの判定は、最小の差分絶対値和をテンプレートのアクティビティで除算して得られる値を閾値と比較することで行うようにしてもよい。この場合におけるアクティビティは、後述する図40のアクティビティ算出部602により、図41のステップS603において算出された値を用いることができる。

**【0161】**

あるいはまた、今回の最小の差分絶対値和を1フレーム前における最小の差分絶対値和で除算することで得られた値を所定の閾値と比較することで、通常処理への復帰が可能であるか否かを判定するようにしてもよい。この場合、アクティビティの計算が不要となる。すなわち、ステップS404では、テンプレートとテンプレート探索範囲の相関が演算され、相関値と閾値の比較に基づいて判定が行われる。

**【0162】**

ステップS404において、通常処理への復帰が可能ではないと判定された場合、ステップS405に進み、継続判定処理が実行される。継続判定処理の詳細は、図39のフローチャートを参照して後述するが、これにより、追尾処理が継続可能であるか否かの判定が行われる。

**【0163】**

ステップS406において、制御部59は、追尾点の追尾が継続可能であるか否かを継続判定処理の結果に基づいて（後述する図39のステップS466、S468で設定されたフラグに基づいて）判定する。追尾点の追尾処理が継続可能である場合には、処理はステップS302に戻り、それ以降の処理が繰り返し実行される。すなわち、追尾点が再び出現するまで待機する処理が繰り返し実行される。

## 【0164】

これに対して、ステップS406において、追尾点の追尾処理が継続可能ではないと判定された場合（後述する図39のステップS465で、追尾点が消失した後の経過フレーム数が閾値THfr以上と判定されるか、または、ステップS467でシーンチェンジ数が閾値THsc以上と判定された場合）、最早、追尾処理は不可能として、追尾処理は終了される。

## 【0165】

図39は、図35のステップS405における継続判定処理の詳細を表している。ステップS461において、制御部59は、変数としての経過フレーム数に1を加算する処理を実行する。経過フレーム数は、図35のステップS401の例外処理の初期化処理（図36のステップS426）において、予め0にリセットされている。

## 【0166】

次にステップS462において、制御部59は、シーンチェンジがあるか否かを判定する。シーンチェンジがあるか否かは、シーンチェンジ検出部53が、常にその検出処理を実行しており、その検出結果に基づいて判定が可能である。シーンチェンジがある場合には、ステップS463に進み、制御部59は変数としてのシーンチェンジ数に1を加算する。このシーンチェンジ数も、図36のステップS426の初期化処理において0にリセットされている。通常処理から例外処理への移行時にシーンチェンジが発生していない場合には、ステップS463の処理はスキップされる。

## 【0167】

次に、ステップS464において、制御部59は、現在設定されているモードがシーンチェンジであるか否かを判定する。このモードは、図36のステップS422、S423において設定されたものである。現在設定されているモードがシーンチェンジである場合には、ステップS467に進み、制御部59は、シーンチェンジ数が予め設定されている閾値THscより小さいか否かを判定する。シーンチェンジ数が閾値THscより小さい場合には、ステップS466に進み、制御部59は継続可のフラグを設定し、シーンチェンジ数が閾値THsc以上である場合には、ステップS468に進み、継続不可のフラグを設定する。

## 【0168】

一方、ステップS464において、モードがシーンチェンジではないと判定された場合（モードがその他であると判定された場合）、ステップS465に進み、制御部59は、経過フレーム数が閾値THfrより小さいか否かを判定する。この経過フレーム数も、図36の例外処理の初期化処理のステップS426において、予め0にリセットされている。経過フレーム数が閾値THfrより小さいと判定された場合には、ステップS466において、継続可のフラグが設定され、経過フレーム数が閾値THfr以上であると判定された場合には、ステップS468において、継続不可のフラグが設定される。

## 【0169】

このように、テンプレートマッチング処理時におけるシーンチェンジ数が閾値THsc以上になるか、または経過フレーム数が閾値THfr以上になった場合には、それ以上の追尾処理は不可能とされる。

## 【0170】

なお、モードがその他である場合には、シーンチェンジ数が0であるという条件も加えて、継続が可能であるか否かを判定するようにしてもよい。

## 【0171】

以上においては、画像のフレームを処理単位とし、すべてのフレームを用いることを前提としたが、フィールド単位で処理したり、すべてのフレームまたはフィールドを利用するのではなく、所定の間隔で間引いて抽出されたフレームまたはフィールドを用いるようにすることも可能である。

## 【0172】

次に、図40を参照して、図5の動き推定部52の構成例について説明する。この実施の形態においては、入力画像が、評価値算出部601、アクティビティ算出部602、お

よび動きベクトル検出部606に供給されている。評価値算出部601は、動きベクトルにより対応付けられる両対象の一致度に関する評価値を算出し、正規化処理部604に供給する。アクティビティ算出部602は、入力画像のアクティビティを算出し、閾値判定部603と正規化処理部604に供給する。動きベクトル検出部606は、入力画像から動きベクトルを検出し、評価値算出部601と統合処理部605に供給する。

#### 【0173】

正規化処理部604は、評価値算出部601より供給された評価値を、アクティビティ算出部602より供給されたアクティビティに基づいて正規化し、得られた値を統合処理部605に供給する。閾値判定部603は、アクティビティ算出部602より供給されたアクティビティを所定の閾値と比較し、その判定結果を統合処理部605に供給する。統合処理部605は、正規化処理部604から供給された正規化情報と、閾値判定部603より供給された判定結果に基づいて、動きベクトルの確度を演算し、得られた確度を動きベクトル検出部606より供給された動きベクトルとともに出力する。

#### 【0174】

次に、図41のフローチャートを参照して、動き推定部52の動き推定処理について説明する。動きベクトルは、点に対するものとして求められているが、その確度は、動きベクトルにより対応付けられる2つの点の近傍の、例えば点を中心とする、小ブロックの画像データを用いて計算される。ステップS601において、動きベクトル検出部606は、入力画像から動きベクトルを検出する。この検出には、例えばブロックマッチング方式や勾配法が用いられる。検出された動きベクトルは、評価値算出部601と統合処理部605に供給される。

#### 【0175】

ステップS602において、評価値算出部601は評価値を算出する。具体的には、例えば、動きベクトルで対応付けられる2つの点を中心とする2つのブロックの画素値の差分絶対値和が算出される。すなわち、ステップS601で動きベクトル検出部606により検出された動きベクトル $V(v_x, v_y)$ と、それに基づく時間的に前のフレームの画像 $F_i$ 上の点 $P(X_p, Y_p)$ 、並びに時間的に後のフレームの画像 $F_j$ 上の点 $Q(X_q, Y_q)$ の関係は次式で表される。

$$Q(X_q, Y_q) = P(X_p, Y_p) + V(v_x, v_y) \quad \dots (1)$$

#### 【0176】

評価値算出部601は点Pを中心とするブロックと、点Qを中心とするブロックについて、次式に基づいて評価値 $Eval(P, Q, i, j)$ を演算する。

#### 【0177】

【数1】

$$Eval(P, Q, i, j) = \sum \sum |F_j(X_q+x, Y_q+y) - F_i(X_p+x, Y_p+y)| \quad \dots (2)$$

#### 【0178】

各ブロックは、1辺が $2L+1$ 画素の正方形とされている。上記式における総和 $\sum \sum$ は、 $x$ が $-L$ から $L$ について、 $y$ が $-L$ から $L$ について、対応する画素同士で行われる。従って、例えば、 $L=2$ である場合、9個の差分が得られ、その絶対値の総和が演算される。評価値は、その値が0に近づくほど、2つのブロックがよく一致していることを表している。

#### 【0179】

評価値算出部601は、生成した評価値を正規化処理部604に供給する。

#### 【0180】

ステップS603において、アクティビティ算出部602は、入力画像からアクティビティを算出する。アクティビティは、画像の複雑さを表す特徴量であり、図42に示されるように、各画素毎に注目画素 $Y(x, y)$ と、それに隣接する8画素 $Y(x+i, y+j)$

j) との差分絶対値和の平均値が、注目画素位置のアクティビティ Activity(x, y) として次式に基づいて演算される。

【0181】

【数2】

$$\text{Activity}(x, y) = \frac{\sum_{j=-1}^1 \sum_{i=-1}^1 |Y(x+i, y+j) - Y(x, y)|}{8} \quad \dots (3)$$

【0182】

図42の例の場合、3×3画素のうち、中央に位置する注目画素 Y(x, y) の値は110であり、それに隣接する8個の画素の値は、それぞれ80, 70, 75, 100, 100, 100, 80, 80であるから、アクティビティ Activity(x, y) は次式で表される。

【0183】

$$\begin{aligned} \text{Activity}(x, y) &= \frac{|80-110| + |70-110| + |75-110| + |100-110|}{8} \\ &\quad + \frac{|100-110| + |100-110| + |80-110| + |80-110|}{8} \\ &= 24.375 \end{aligned}$$

となる。

【0184】

同様の処理が、そのフレームのすべての画素について実行される。

【0185】

ブロック単位で動きベクトル確度を算出するため、次式で表されるブロック内の全画素のアクティビティの総和が、そのブロックのアクティビティ (ブロックアクティビティ) Blockactivity(i, j) と定義される。

【0186】

【数3】

$$\text{Blockactivity}(i, j) = \sum \sum |\text{Activity}(x, y)| \quad \dots (4)$$

【0187】

なお、アクティビティとしては、この他、分散値、ダイナミックレンジなどとする事も可能である。

【0188】

閾値判定部603は、ステップS604において、アクティビティ算出部602により算出されたブロックアクティビティを予め設定されている所定の閾値と比較する。そして、入力されたブロックアクティビティが閾値より大きいかな否かを表すフラグを統合処理部605に出力する。

【0189】

具体的には、実験の結果、ブロックアクティビティと評価値は、動きベクトルをパラメータとして、図43に示される関係を有する。図43において、横軸はブロックアクティビティ Blockactivity(i, j) を表し、縦軸は評価値 Eval を表している。動きが正しく検出されている場合 (正しい動きベクトルが与えられている場合)、そのブロックアクティビティと評価値の値は、曲線621より図中下側の領域R1に分布する。これに対して誤った動き (不正解の動きベクトル) が与えられた場合、そのブロックアクティビティと評価値の値は、曲線622より、図中左側の領域R2に分布する (曲線622より上側の領域R2以外の領域と曲線621より下側の領域R1以外の領域には殆ど分布がない)。曲線621と曲線622は、点Pにおいて交差する。この点Pにおけるブロックアクティビティの値が閾値Thaとされる。閾値Thaは、ブロックアクティビティの値がそれより小さい場合

には、対応する動きベクトルが正しくない可能性があることを意味する（この点については後に詳述する）。閾値判定部603は、アクティビティ算出部602より入力されたブロックアクティビティの値が、この閾値THaより大きいかな否かを表すフラグを統合処理ブロック605に出力する。

**【0190】**

ステップS605において、正規化処理部604は、正規化処理を実行する。具体的には、正規化処理部604は、次式に従って動きベクトル確度VCを演算する。

**【0191】**

$$VC = 1 - \text{評価値} / \text{ブロックアクティビティ} \quad \dots (5)$$

**【0192】**

但し、動きベクトル確度VCの値が0未満となる場合にはその値を0に置き換える。動きベクトル確度VCのうち、評価値をブロックアクティビティで割り算して得られた値は、その値によって規定される図43のグラフ上の位置が、原点Oと点Pを結ぶ傾きが1の直線623より、図中下側の領域内であるのか、図中上側の領域内であるのかを表す。すなわち、直線623の傾きは1であり、評価値をブロックアクティビティで割り算して得られた値が1より大きければ、その値に対応する点は、直線623の上側の領域に分布する点であることを意味する。そしてこの値を1から減算して得られる動きベクトル確度VCは、その値が小さい程、対応する点が領域R2に分布する可能性が高いことを意味する。

**【0193】**

これに対して、評価値をブロックアクティビティで割り算して得られた値が1より小さければ、その値に対応する点は、直線623の図中下側の領域に分布することを意味する。そして、そのときの動きベクトル確度VCは、その値が大きい程（0に近い程）、対応する点が領域R1に分布することを意味する。正規化処理部604は、このようにして演算して得られた動きベクトル確度VCを統合処理部605に出力する。

**【0194】**

ステップS606において、統合処理部605は、統合処理を実行する。この統合処理の詳細は、図44のフローチャートに示されている。

**【0195】**

統合処理部605は、ステップS631において、ブロックアクティビティが閾値THa以下かな否かを判定する。この判定は、閾値判定部603より供給されたフラグに基づいて行われる。ブロックアクティビティが閾値THa以下である場合には、ステップS632において統合処理部605は、正規化処理部604が算出した動きベクトル確度VCの値を0に設定する。ステップS631において、アクティビティの値が閾値THaより大きいと判定された場合には、ステップS632の処理はスキップされ、正規化処理部604で生成された動きベクトル確度VCの値が、そのまま動きベクトルとともに出力される。

**【0196】**

これは、正規化処理部604において演算された動きベクトルの確度VCの値が正であったとしても、ブロックアクティビティの値が閾値THaより小さい場合には、正しい動きベクトルが得られていない可能性があるからである。すなわち、図43に示されるように、原点Oと点Pの間においては、曲線622が、曲線621より図中下側に（直線623より下側に）突出することになる。ブロックアクティビティの値が閾値THaより小さい区間であって、曲線621と曲線622において囲まれる領域R3においては、評価値をブロックアクティビティで割り算して得られる値は、領域R1とR2の両方に分布し、正しい動きベクトルが得られていない可能性が高い。そこで、このような分布状態である場合には、動きベクトルの確度は低いものとして処理するようにする。このため、ステップS632において、動きベクトル確度VCは、その値が正であったとしても、閾値THaより小さい場合には、0に設定される。このようにすることで、動きベクトル確度VCの値が正である場合には、正しい動きベクトルが得られている場合であることを確実に表すことが可能となる。しかも、動きベクトル確度VCの値が大きい程、正しい動きベクトルが得られている確率が高くなる（分布が領域R1に含まれる確率が高くなる）。

## 【0197】

このことは、一般的に、輝度変化が少ない領域（アクティビティが小さい領域）では信頼性が高い動きベクトルを検出することが困難であるとの経験上の法則とも一致する。

## 【0198】

図45は、図5の背景動き推定部54の構成例を表している。この構成例においては、背景動き推定部54は、頻度分布算出部651と背景動き決定部652により構成されている。

## 【0199】

頻度分布算出部651は、動きベクトルの頻度分布を算出する。ただし、この頻度には、動き推定部12より供給される動きベクトル確度VCを用いることで、確からしい動きに重みが与えられるように、重み付けが行われる。背景動き決定部652は、頻度分布算出部651により算出された頻度分布に基づいて、頻度が最大となる動きを背景動きとして決定する処理を行い、領域推定関連処理部55へ出力する。

## 【0200】

図46を参照して、背景動き推定部54の背景動き推定処理について説明する。

## 【0201】

ステップS651において、頻度分布算出部651は、動き頻度分布を算出する。具体的には、頻度分布算出部651は、背景動きの候補としての動きベクトルのx座標とy座標がそれぞれ基準点から±16画素分の範囲で表されるとすると、 $1089$ 個（ $=16 \times 2 + 1 \times (16 \times 2 + 1)$ ）の箱、すなわち動きベクトルがとり得る値に対応する座標分の箱を用意し、動きベクトルが発生した場合、その動きベクトルに対応する座標に1を加算する。このようにすることで、動きベクトルの頻度分布を算出することができる。

## 【0202】

ただし、1個の動きベクトルが発生した場合、1を加算していくと、確度が低い動きベクトルの発生頻度が多い場合、その確実性が低い動きベクトルが背景動きとして決定されてしまう恐れがある。そこで、頻度分布算出部651は、動きベクトルが発生した場合、その動きベクトルに対応する箱（座標）に、値1を加算するのではなく、値1に動きベクトル確度VCを乗算した値（＝動きベクトル確度VCの値）を加算する。動きベクトル確度VCの値は、0から1の間の値として正規化されており、その値が1に近いほど確度が高い値である。従って、このようにして得られた頻度分布は、動きベクトルをその確度に基づいて重み付けした頻度分布となる。これにより、確度の低い動きが背景動きとして決定される恐れが少なくなる。

## 【0203】

次に、ステップS652において、頻度分布算出部651は、動き頻度分布を算出する処理を全ブロックについて終了したか否かを判定する。まだ処理していないブロックが存在する場合には、ステップS651に戻り、次のブロックについてステップS651の処理が実行される。

## 【0204】

以上のようにして、全画面に対して動き頻度分布算出処理が行われ、ステップS652において、全ブロックの処理が終了したと判定された場合、ステップS653に進み、背景動き決定部652は、頻度分布の最大値を検索する処理を実行する。すなわち、背景動き決定部652は、頻度分布算出部651により算出された頻度の中から最大の頻度のものを選択し、その頻度に対応する動きベクトルを背景動きの動きベクトルとして決定する。この背景動きの動きベクトルは、領域推定関連処理部55に供給され、例えば、図20のステップS204や図23のステップS231の全画面動きと背景動きが一致するか否かの判定処理に用いられる。

## 【0205】

図47は、図5のシーンチェンジ検出部53の詳細な構成例を表している。この例においては、動きベクトル確度平均算出部671と閾値判定部672によりシーンチェンジ検出部53が構成されている。



## 【0206】

動きベクトル確度平均算出部671は、動き推定部52より供給された動きベクトル確度VCの全画面の平均値を算出し、閾値判定部672に出力する。閾値判定部672は、動きベクトル確度平均算出部671より供給された平均値を、予め定められている閾値と比較し、その比較結果に基づいて、シーンチェンジであるか否かを判定し、判定結果を制御部59に出力する。

## 【0207】

次に、図48のフローチャートを参照して、シーンチェンジ検出部53の動作について説明する。ステップS681において、動きベクトル確度平均算出部671は、ベクトル確度の総和を算出する。具体的には、動きベクトル確度平均算出部671は、動き推定部52の統合処理部605より出力された各ブロック毎に算出された動きベクトル確度VCの値を加算する処理を実行する。ステップS682において、動きベクトル確度平均算出部671は、ベクトル確度VCの総和を算出する処理が全ブロックについて終了したか否かを判定し、まだ終了していない場合には、ステップS681の処理を繰り返す。この処理を繰り返すことで、1画面分の各ブロックの動きベクトル確度VCの総和が算出される。ステップS682において1画面全部についての動きベクトル確度VCの総和の算出処理が終了したと判定された場合、ステップS683に進み、動きベクトル確度平均算出部671は、ベクトル確度VCの平均値を算出する処理を実行する。具体的には、ステップS681の処理で算出された1画面分のベクトル確度VCの総和を、足し込まれたブロック数で除算して得られた値が平均値として算出される。

## 【0208】

ステップS684において、閾値判定部672は、ステップS683の処理で動きベクトル確度平均算出部671により算出された動きベクトル確度VCの平均値を、予め設定されている閾値と比較し、閾値より小さいか否かを判定する。一般的に、動画中の時刻が異なる2フレーム間でシーンチェンジが発生すると、対応する画像が存在しないため、動きベクトルを算出しても、その動きベクトルは確からしくなることになる。そこで、ベクトル確度VCの平均値が閾値より小さい場合には、ステップS685において、閾値判定部672はシーンチェンジフラグをオンし、閾値より小さくない場合（閾値以上である場合）、ステップS586において、シーンチェンジフラグをオフにする。シーンチェンジフラグのオンは、シーンチェンジがあったことを表し、そのオフは、シーンチェンジが無いことを表す。

## 【0209】

このシーンチェンジフラグは、制御部59へ供給され、図36のステップS421におけるシーンチェンジの有無の判定、並びに図39のステップS462のシーンチェンジの有無の判定に利用される。

## 【0210】

以上のように、図1のオブジェクト追尾部26を構成することにより、追尾すべきオブジェクト51（図3）が回転したり、オクルージョンが発生したり、あるいはシーンチェンジにより、オブジェクト51の追尾点51Aが一時的に表示されなくなるような場合でも、画像の中で移動するオブジェクト51（追尾点51A）を正確に追尾することができる。

## 【0211】

このようにして追尾されるオブジェクト51の追尾点51Aの位置情報が、図1のオブジェクト追尾部26による追尾結果としてエリア設定部25に出力されることにより、エリア設定部25によって、上述したように補正対象エリア52が設定される。そして、画像補正部22が補正対象エリア52の中の画像のぼけ（フォーカスぼけ）を除去する。

## 【0212】

次に、図1の画像補正部22の詳細な構成例と、その動作について説明する。図49は、画像補正部22の詳細な構成例を示すブロック図である。この例では画像補正部22に、エリア設定部25の出力信号に基づいて制御信号を生成し、その制御信号を各部に供給

する制御信号生成部 741、入力画像の特徴を検出する画像特徴検出部 742、制御信号に基づいて、アドレスの演算を行うアドレス演算部 743、アドレス演算部 743により演算されたアドレスに基づいて、予め記憶された所定の係数を出力する係数 ROM 744、および入力画像の中の所定の領域に対応する複数の画素を抽出する領域抽出部 745 が設けられている。

#### 【0213】

また、領域抽出部 745 から出力された画素のレベルに対して、係数 ROM 744 から出力された係数に基づく積和演算を行い、新たに修正された画素レベルを出力する積和演算部 746、および積和演算部 746 の出力結果と制御信号に基づいて、補正対象エリア 52 内の画像と、背景 53 を合成し、出力する画像合成部 747 が設けられている。

#### 【0214】

図 50 は、制御信号生成部 741 が生成する制御信号の例を示す図である。制御信号 A は、入力画像の中の修正すべき部分（補正対象エリア 52）を特定する信号であり、エリア設定部 25 の出力に基づいて生成され、領域抽出部 745 と画像合成部 747 に供給される。制御信号 B は、後述するぼけの度合いを表すパラメータ  $\sigma$  を特定する信号であり、アドレス演算部 743 に供給される。パラメータ  $\sigma$  の値は、例えば、制御部 27 を介して行われる、ユーザの指定に基づいて特定されるようにしてもよいし、予め設定されるようにしてもよい。

#### 【0215】

制御信号 C は、後述するぼけのモデル式を解くために用いられる関係式の重み  $W_a$  の切り替えを指定する信号であり、アドレス演算部 743 に供給される。制御信号 D は、画像の特徴を検出するとき用いられる閾値の切り替えを指定する信号であり、画像特徴検出部 742 に供給される。制御信号 C と D については、監視カメラシステム 1 の特性などを考慮して予め設定されるようにしてもよいし、制御部 27 を介して行われる、ユーザの指定に基づいて生成されるようにしてもよい。

#### 【0216】

次に、画像のぼけの原理について説明する。いま、カメラのピントが適正に設定され被写体がフォーカスぼけしていない画像の画素のレベル  $X$  を真値とし、カメラのピントが外れて被写体がフォーカスぼけした画像の画素のレベル  $Y$  を観測値とする。画像を構成する複数の画素を表現するために、画像の水平方向の座標を  $x$  であらわし、垂直方向の座標を  $y$  で表すと、真値は、 $X(x, y)$  で表され、観測値は、 $Y(x, y)$  で表すことができる。

#### 【0217】

本発明では、ぼけのモデル式として、式 (6) を適用する。式 (6) においては、式 (7) に示されるガウス関数を用い、真値  $X(x, y)$  にガウス関数を畳み込むことにより観測値  $Y(x, y)$  が得られる。

#### 【0218】

##### 【数 4】

$$Y(x, y) = \sum_{\substack{-r < i < r \\ -r < j < r}} [W(i, j) \times X(x+i, y+j)] \quad \dots (6)$$

#### 【0219】

##### 【数 5】

$$W(j, i) = \frac{1}{2\pi\sigma^2} e^{-\frac{j^2+i^2}{2\sigma^2}} \quad \dots (7)$$

#### 【0220】

式 (6) において、パラメータ  $\sigma$  は、ぼけの度合いを表すパラメータである。

#### 【0221】

式(6)によれば、1つの観測値 $Y(x, y)$ は、変数 $i$ と $j$  ( $-r < i < r$ ,  $-r < j < r$ )により変化する複数の真値 $X(x+i, y+j)$ を係数 $W$ で重みづけすることにより求められる。従って、ぼけのない画像の1つの画素のレベルは、ぼけた画像の複数の画素のレベルに基づいて得られたものとされる。

#### 【0222】

また、上述したパラメータ $\sigma$ の値により画像のぼけの度合いが変化する。パラメータ $\sigma$ の値が比較的小さい場合、真値の情報が観測値において広範囲に拡散されておらず、比較的ぼけの小さい画像となる。これに対して、パラメータ $\sigma$ の値が比較的大きい場合、真値の情報が観測値において広範囲に拡散され、比較的ぼけの大きい画像となる。

#### 【0223】

このように、パラメータ $\sigma$ の値の変化により、画像のぼけの度合いは変化する。このため、画像のぼけを正確に修正するためには、パラメータ $\sigma$ の値を適切に求める必要がある。本発明においては、パラメータ $\sigma$ の値をユーザが指定する。あるいは、監視カメラシステム1の特性などが考慮され、最適な値が予め設定されるようにしてもよい。

#### 【0224】

図51乃至図54を参照して、画像のぼけの原理についてさらに詳しく説明する。図51Aは、簡単のため、画素が水平方向に一次元に配列されたものとして、ある画像における、真値 $X_0$ 乃至 $X_8$ を表す図である。図51Cは、図51Aに対応する観測値を表す図である。図51Bは、係数 $W(i)$ の大きさを棒グラフ状に表した図である。この例では、変数 $i$ が、 $-2 < i < 2$ とされ、中央の棒グラフが、係数 $W(0)$ とされ、左端の棒グラフから順番に係数 $W(-2)$ ,  $W(-1)$ ,  $W(0)$ ,  $W(1)$ ,  $W(2)$ とされる。

#### 【0225】

ここで、式(6)に基づいて、図51Cの観測値 $Y_2$ を求めると、次のようになる。

#### 【0226】

$$Y_2 = W(-2)X_2 + W(-1)X_3 + W(0)X_4 + W(1)X_5 + W(2)X_6$$

#### 【0227】

同様に、図51Cの観測値 $Y_0$ を求める場合、真値の中で、図52の枠790-1で示される部分に基づいて、演算を行うことにより、次のように観測値 $Y_0$ が求められる。

#### 【0228】

$$Y_0 = W(-2)X_0 + W(-1)X_1 + W(0)X_2 + W(1)X_3 + W(2)X_4$$

#### 【0229】

さらに、観測値 $Y_1$ を求める場合、図52の枠790-2で示される部分に基づいて、演算を行うことにより、次のように観測値 $Y_1$ が求められる。

#### 【0230】

$$Y_1 = W(-2)X_1 + W(-1)X_2 + W(0)X_3 + W(1)X_4 + W(2)X_5$$

#### 【0231】

$Y_3$ ,  $Y_4$ についても、同様にして求めることができる。

#### 【0232】

図53と図54は、図51Aと図51Cの関係を2次元で表したものである。すなわち、図53を構成する各画素のレベルは、観測値であり、図54を構成する各画素のレベルを真値として、得られたものである。この場合、図53における画素Aに対応する観測値 $Y(x, y)$ は次のようにして求められる。

#### 【0233】

$$(Y(x, y) = W(-2, -2)X(x-2, y-2) + W(-1, -2)X(x-1, y-2) \\ + W(0, 2)X(x, y-2) \dots + W(2, 2)X(x+2, y+2))$$

#### 【0234】

すなわち、図53の画素Aに対応する観測値は、図54において画素A' (画素Aに対応する)を中心として枠aで示される25 (=5×5)個の画素に対応する真値に基づいて求められる。同様に、図53の画素B (画素Aの図中右隣の画素)に対応する観測値は

、図 5 4 において、画素 B' (画素 B に対応する) を中心とした 25 個の画素に対応する真値に基づいて求められ、図 5 3 の画素 C に対応する観測値は、図 5 4 において、画素 C' (画素 C に対応する) を中心とした 25 個の画素に対応する真値に基づいて求められる。図 5 3 の画素 B と C に対応する観測値  $Y(x+1, y)$  と  $Y(x+2, y)$  を求める式を次に示す。

【0 2 3 5】

$$Y(x+1, y) = W(-2, -2)X(x-1, y-2) + W(-1, -2)X(x, y-2) + W(0, -2)X(x+1, y-2) \dots + W(2, 2)X(x+3, y+2)$$

【0 2 3 6】

$$Y(x+2, y) = W(-2, -2)X(x, y-2) + W(-1, -2)X(x+1, y-2) + W(0, -2)X(x+2, y-2) \dots + W(2, 2)X(x+4, y+2)$$

【0 2 3 7】

このようにして、図 5 3 の各画素に対応する観測値をもとめていくと、式 (8) 乃至 (11) に示されるような行列式が得られる

【0 2 3 8】

【数 6】

$$Y_f = \begin{pmatrix} Y(x, y) \\ Y(x+1, y) \\ Y(x+2, y) \\ Y(x+3, y) \\ \vdots \\ Y(x, y+1) \\ Y(x+1, y+1) \\ \vdots \\ Y(x+7, y+7) \end{pmatrix} \dots (8)$$

【0 2 3 9】

【数 7】

$$W_f = \begin{pmatrix} W(-2, -2) & W(-1, -2) & \dots & W(2, 2) \\ W(-2, -2) & W(-1, -2) & \dots & W(2, 2) \\ \vdots & \vdots & \vdots & \vdots \\ W(-2, -2) & W(-1, y-2) & \dots & W(2, 2) \end{pmatrix} \dots (9)$$

【0 2 4 0】

【数 8】

$$X_f = \begin{pmatrix} X(x-2, y-2) & X(x-1, y-2) & \dots & X(x, y-2) \\ X(x-1, y-2) & X(x, y-2) & \dots & X(x+1, y-2) \\ \vdots & \vdots & \vdots & \vdots \\ X(x+2, y+2) & X(x+3, y+2) & \dots & X(x+9, y+9) \end{pmatrix} \dots (10)$$

【0 2 4 1】

【数 9】

$$Y_f = W_f X_f \dots (11)$$

【0 2 4 2】

ここで、式(11)に示した行列式において、行列 $W_f$ の逆行列を求めることができれば、観測値 $Y_f$ に基づいて真値 $X_f$ を求めることができる。すなわち、ぼけた画像の画素に基づいて、ぼけない画像の画素を得ることができ、ぼけた画像を修正することができる。

【0243】

しかし、式(8)乃至式(11)に示した行列式は、図51乃至図54を参照して上述したように、観測値の画素に対して、真値の画素が多く、このままでは逆行列を求めることができない(例えば、図52の例では、観測値の画素1個に対して真値の画素5個が必要となる)。

【0244】

そこで、式(8)乃至式(11)に加えて、式(12)乃至式(15)に示される関係式を導入する。

【0245】

【数10】

$$W_a(p_1)W_1(p_2)(X(x, y) - X(x, y-1)) = 0 \quad \dots (12)$$

【0246】

【数11】

$$W_a(p_1)W_2(p_2)(X(x, y) - X(x+1, y)) = 0 \quad \dots (13)$$

【0247】

【数12】

$$W_a(p_1)W_3(p_2)(X(x, y) - X(x, y+1)) = 0 \quad \dots (14)$$

【0248】

【数13】

$$W_a(p_1)W_4(p_2)(X(x, y) - X(x-1, y)) = 0 \quad \dots (15)$$

【0249】

式(12)乃至式(15)は、隣接する画素のレベルの差分について限定を加えるものであり、求めるべき真値が、画像の平坦な(隣接する画素のレベルと大きな差がない)部分である場合には矛盾がない。しかし、求めるべき真値がエッジ部分である(隣接する画素のレベルと大きな差がある)場合には矛盾が生じ、修正した画像に劣化が生じる恐れがある。このため、ぼけた画像を適正に修正するためには、式(12)乃至式(15)の4つの関係式を、真値のエッジ部分をまたがないように画素ごとに使い分ける必要がある。

【0250】

そこで、画像特徴検出部742において、入力画像の中のエッジ部分と平坦部分の判定を行い、どの方向(例えば、上下左右)に平坦になっているかを表すコード $p_2$ を生成する。なお、画像特徴検出部742の詳細な動作については、図59を参照して後述する。また、本発明では、入力画像(観測値)の中のエッジ部分と平坦部分の判定結果が、真値の中でのエッジ部分と平坦部分の判定結果に等しいと仮定する。

【0251】

式(12)乃至式(15)において、コード $p_2$ の関数である関数 $W_1$ 乃至 $W_4$ は、重み関数とされる。本発明においては、コード $p_2$ に応じてこの重み関数 $W_1$ 乃至 $W_4$ が制御されることで、画素ごとの関係式の使い分けが行われるようにする。図55にコード $p_2$ に対応する重み関数 $W_1$ 乃至 $W_4$ の値を示す。この重み関数の値が大きい場合、式(12)乃至式(15)において平坦であるという意味合いが強くなり、重み関数の値が小さい場合、その意味合いが弱くなる(エッジである意味合いが強くなる)。

【0252】

コード $p_2$ は、4ビットにより構成されており、それぞれのビットは、左から順番に、上

、右、下または左方向に平坦か否かを示しており、その方向に平坦である場合には、対応するビットが「1」に設定される。例えば、コードp2が「0001」の場合、注目画素の左方向に平坦であり、それ以外の方向は平坦ではない（エッジが存在する）ことを表す。このため、コードp2が「0001」の場合、重み関数W4の値が大きくなり、式（12）乃至式（15）の4つの関係式の中で式（15）の重みが大きくなる。このようにすることで、コードp2により、4つの関係式の重みを変化させることができ、4つの関係式を、エッジをまたがないように画素ごとに使い分けることができる。

#### 【0253】

例えば、図56に示されるように、注目画素Xaの上方向と左方向が平坦であり、右方向と下方向がエッジである場合、コードp2により、式（12）乃至式（15）の4つの関係式の重みを変化させることにより、隣接する画素のレベルの差分について、「 $Xa - Xb = 0$ 」、「 $Xa - Xc = 0$ 」という限定が加えられるが、「 $Xa - Xd = 0$ 」、「 $Xa - Xe = 0$ 」という限定は加えられない。なお、Xb、Xc、Xd、Xeは、それぞれ注目画素Xaの右、下、上、または左に隣接する画素を表す。

#### 【0254】

また、式（12）乃至式（15）において、関数Waは、別の重み関数であり、やはりコードp1により重み関数Waの値が変化する。重み関数Waの値を変化させることで、修正された画像の全体のノイズ、ディテールを制御することができる。重み関数Waの値が大きいと、修正された画像においてノイズの影響が小さく感じられ、ノイズ感が減少する。また、重み関数Waの値が小さいと、修正された画像においてディテールが強調されたように感じられ、ディテール感が向上する。なお、重み関数Waの値を変化させるコードp1は、図50の制御信号Cに対応している。

#### 【0255】

このように、式（8）乃至式（11）に加えて、式（12）乃至式（15）に示される関係式を導入する。これにより、式（16）に示されるような逆行列を演算することが可能になり、その結果、観測値に基づいて真値を求めることができる。

#### 【0256】

##### 【数14】

$$X_s = W_s^{-1} Y_s \quad \dots (16)$$

#### 【0257】

本発明では、観測値Ysにかかる係数 $W_s^{-1}$ が係数ROM744に予め保持され、領域抽出部745により抽出された入力画像に対して、式（16）の行列式の演算（積和演算）が積和演算部746により行われる。このようにすることで、画像の修正を行う都度、逆行列演算を行う必要がなく、積和演算だけでほけを修正することが可能になる。ただし、入力画像に応じて、パラメータ $\sigma$ や、上述した4つの関係式が異なるため、想定しうるそれらの全ての組み合わせでの逆行列演算を予め行っておき、パラメータ $\sigma$ 、コードp2などに対応するアドレスを定めて、そのアドレス毎に異なる係数が係数ROM744に格納される。

#### 【0258】

しかし、例えば、図54に示される枠(t)内の25（=5×5）個のすべてに画素において、重み関数W1乃至W4の組み合わせを変化させ、4つの関係式を切り替えた場合、15（=図55に示した関数W1乃至W4の組み合わせ）の25（枠(t)内の画素数）乗の組み合わせが存在し、それぞれの組み合わせごとに逆行列演算を行うと、係数の数が膨大になり、係数ROM744の容量には制限があるため、全ての係数を格納しきれなくなるおそれがある。このような場合、枠(t)内の中心画素であるXtのみ、その特徴に基づいて、コードp2を変化させて関係式を切り替え、枠(t)内の画素Xt以外画素の関係式については、例えばコードp2が擬似的に、「1111」に固定されるようにしてもよい。このようにすることで、係数の組み合わせを15通りに限定することができる。

## 【0259】

なお以上においては、ぼけの原理（モデル式）を説明するために、ガウス関数の定義域を、 $-2 \leq (x, y) \leq 2$  としたが、実際には、パラメータ  $\sigma$  の値が十分大きくても対応できるような範囲が設定される。また、式（12）乃至式（15）に示した関係式についても、画像の特徴を記述する式であれば、これに限定されるものではない。さらに、係数ROM 744の容量に制限がある場合の例として、ぼけの中心位相（Xt）のみに限定して関係式を切り替える例を説明したが、それに限定されるものではなく、係数ROM 744の容量に応じて、関係式の切り替え方法を変えてもよい。

## 【0260】

次に図57を参照して、画像補正部22によるぼけ修正処理について説明する。ステップS801において、画像補正部22は、処理対象領域を検出する。この処理対象領域は、ぼけの修正を行うべき領域、すなわち補正対象エリア52であり、エリア設定部25から出力される信号に基づいて検出される。

## 【0261】

ステップS802において、画像補正部22は、パラメータ  $\sigma$  の値を取得する。パラメータ  $\sigma$  は、ユーザにより指定されるようにしてもよいし、予め設定された値が取得されるようにしてもよい。ステップS803において、画像補正部22は、図58を参照して後述する画像補正処理を実行する。これにより、ぼけた画像が修正され、出力される。

## 【0262】

このようにして、補正対象エリア52内の画像については、画像のぼけが除去されて鮮明な画像となる。

## 【0263】

次に、図58を参照して、図57のステップS803の画像補正処理の詳細について説明する。

## 【0264】

ステップS821において、画像特徴検出部742は、図59を参照して後述する画像特徴検出処理を実行する。これにより、注目画素に対して、どの方向に平坦なのかが判定され、図55を参照して上述したコードp2が生成され、アドレス演算部743に出力される。

## 【0265】

ステップS822において、アドレス演算部743は、係数ROM 744のアドレスを演算する。係数ROM 744のアドレスは、例えば、コードp2に対応する4ビット（画像特徴検出部742の出力）、パラメータ  $\sigma$  の値を表す4ビット（図50の制御信号B）、および、上述した4つの関係式の重み関数Waを切り替えるコードp1に対応する2ビット（図50の制御信号C）により構成され、0乃至1023の1024（2の10乗）個アドレスが存在する。アドレス演算部743は、画像特徴検出部742の出力、制御信号B、および制御信号Cに基づいて、対応するアドレスを演算する。

## 【0266】

ステップS823において、アドレス演算部743は、ステップS822で演算したアドレスに基づいて、係数ROM 744から係数を読み出し、積和演算部746に供給する。

## 【0267】

ステップS824において、積和演算部746は、ステップS823で読み出された係数に基づいて、画素毎に積和演算を行い、その結果を後処理部747に出力する。これにより、上述したように、観測値から真値が求められ、ぼけた画像が修正される。

## 【0268】

ステップS825において、画像合成部747は、図62を参照して後述する画像合成処理を実行する。これにより、画素毎に、積和演算部746の処理結果を出力するか、入力画像をそのまま出力するかが判定される。ステップS826において、後処理部747は、補正後処理され、選択された画像を出力する。

## 【0269】

次に図59を参照して、図58のステップS821の画像特徴検出処理について説明する。ステップS841において、画像特徴検出部742は、ブロックを抽出し、ステップS842において、ステップS841で抽出されたブロック間の差分を演算する（その詳細は、図61を参照して後述する）。ステップS843において、画像特徴検出部742は、ステップS842で演算されたブロック差分を予め設定されている閾値と比較し、その比較結果に基づいて、ステップS844において、注目画素に対して平坦な方向を表すコードであるコードp2を出力する。

## 【0270】

図60と図61を参照して、画像特徴検出処理について、さらに詳しく説明する。図60は、画像特徴検出部742の詳細な構成例を示すブロック図である。同図の左側には、入力された画像の中から所定のブロックを抽出するブロック切り出し部841-1乃至841-5が設けられている。ブロック切り出し部841-1乃至841-5は、例えば、図61A乃至図61Eに示されるように、図中黒い丸で示される注目画素（いま修正すべき画素）の周辺の、注目画素を含む9（=3×3）個の画素で構成される5つのブロックを抽出する。

## 【0271】

図61Aに示されるブロック881は、その中心に注目画素を有する中心ブロックであり、ブロック切り出し部841-5により抽出される。図61Bに示されるブロック882は、ブロック881を画素1個分図中上に移動した上側ブロックであり、ブロック切り出し部841-3により抽出される。図61Cに示されるブロック883は、ブロック881を画素1個分図中左に移動した左側ブロックであり、ブロック切り出し部841-4により抽出される。

## 【0272】

図61Dに示されるブロック884は、ブロック881を画素1個分図中下に移動した下側ブロックであり、ブロック切り出し部841-1により抽出される。図61Eに示されるブロック885は、ブロック881を画素1個分図中右に移動した右側ブロックであり、ブロック切り出し部841-2により抽出される。ステップS841においては、注目画素毎に、ブロック881乃至885の5つのブロックが抽出される。

## 【0273】

ブロック切り出し部841-1乃至841-5により抽出された各ブロックを構成する画素の情報は、ブロック差分演算部842-1乃至842-4に出力される。ブロック差分演算部842-1乃至842-4は、各ブロックの画素の差分を、例えば、次のようにして演算する。

## 【0274】

いま、ブロック881の9個の画素のうち一番上の行の3個の画素（のレベル）を左からa(881), b(881), c(881)とする。中央の行の3個の画素を左からd(881), e(881), f(881)とする。一番下の行の3個の画素を左からg(881), h(881), i(881)とする。同様に、ブロック884の9個の画素についても、一番上の行の3個の画素（のレベル）を左からa(884), b(884), c(884)とし、中央の行の3個の画素を左からd(884), e(884), f(884)とし、一番下の行の3個の画素を左からg(884), h(884), i(884)とする。ブロック差分演算部842-1は、ブロック差分B(1)を次のように演算する。

## 【0275】

$$B(1) = |a(881) - a(884)| + |b(881) - b(884)| + |c(881) - c(884)| + \dots + |i(881) - i(884)|$$

## 【0276】

すなわち、ブロック差分B(1)は、ブロック881（中心）とブロック884（下）において対応する各画素のレベルの差分の絶対値の総和である。同様にして、ブロック差分演算部842-2は、ブロック881（中心）とブロック885（右）において対応する



各画素のレベルの差分の絶対値の総和を求め、ブロック差分B(2)を演算する。さらに、ブロック差分演算部842-3は、ブロック881(中心)とブロック882(上)について、ブロック差分演算部842-3は、ブロック881(中心)とブロック883(左)について、それぞれ対応する各画素のレベルの差分の絶対値の総和を求め、ブロック差分B(3)とB(4)を演算する。

#### 【0277】

ステップS842においては、このように中心ブロックと上下左右の4方向のブロックとの差分であるブロック差分B(1)乃至B(4)が演算され、その結果は、対応する閾値判定部843-1乃至843-4にそれぞれ出力されると同時に、最小方向判定部844にも供給される。

#### 【0278】

閾値判定部843-1乃至843-4は、それぞれブロック差分B(1)乃至B(4)を予め設定された閾値と比較し、その大小を判定する。なお、この閾値は制御信号Dに基づいて切り替えられる。閾値判定部843-1乃至843-4は、それぞれブロック差分B(1)乃至B(4)が予め設定された閾値より大きい場合、その方向はエッジ部分であると判定し、「0」を出力し、閾値より小さい場合、その方向は平坦な部分であると判定し、「1」を出力する。

#### 【0279】

ステップS843においては、このようにしてブロック差分と閾値の比較が行われる。閾値判定部843-1乃至843-4の出力結果は、4ビットのコードとしてセレクタ845に出力される。例えば、ブロック差分B(1)、B(3)およびB(4)が閾値より小さく、ブロック差分B(2)が閾値より大きい場合、コードとして「1011」が出力される。

#### 【0280】

ところで、ブロック差分B(1)乃至B(4)が、全て閾値より大きくなってしまう場合(平坦な部分がない場合)も考えられる。この場合、閾値判定部843-1乃至843-4から、コードとして「0000」が出力される。しかし、図55に示されるように、コードp2が「0000」の場合、対応する重み関数W1乃至W4が特定できない。そこで、セレクタ845は、閾値判定部843-1乃至843-4からの出力結果が「0000」か否かを判定し、閾値判定部843-1乃至843-4からの出力結果が「0000」であると判定された場合、最小方向判定部844からの出力をコードp2として出力する。

#### 【0281】

最小方向判定部844は、ブロック差分B(1)乃至B(4)の中で、最小の値を判定し、判定結果に対応した4ビットのコードを、閾値判定部843-1乃至843-4がコードを出力するのと同じタイミングで、セレクタ845に出力する。例えば、ブロック差分B(1)乃至B(4)の中で、B(1)が最小であると判定された場合、最小方向判定部844は、コードとして「1000」をセレクタ845に出力する。

#### 【0282】

このようにすることで、閾値判定部843-1乃至843-4からコード「0000」が出力されても、最小方向判定部844から出力されたコード「1000」がコードp2として出力されるようにすることができる。勿論、閾値判定部843-1乃至843-4からの出力結果が「0000」ではない場合は、閾値判定部843-1乃至843-4からの出力結果がコードp2として出力される。ステップS844においては、このようにしてコードp2が生成され、アドレス演算部743に出力される。

#### 【0283】

次に、図62を参照して、図58のステップS825の画像合成処理について説明する。ステップS861において、画像合成部747は、積和演算部746からの出力結果に基づいて、画素の分散度を演算する。これにより、注目画素の周囲の画素の分散度合いが演算される。ステップS862において、画像合成部747は、ステップS862で演算された分散度は予め設定された閾値より大きいかな否かを判定する。

## 【0284】

ステップS862において、分散度が閾値より大きいと判定された場合、画像合成部747は、ステップS863において、注目画素に対応する入力画像入れ替えフラグをONに設定する。一方、分散度が閾値より大きくないと判定された場合、画像合成部747は、ステップS64において、注目画素に対応する入力画像入れ替えフラグをOFFに設定する。

## 【0285】

入力画像において、もともとぼけていない部分の画素に対して、積和演算部746により積和演算を行うと、その画素の周囲の画像のアクティビティが大きくなり、かえって画像が劣化してしまう場合がある。ここで、分散度が閾値より大きい場合、その画素は劣化した画素であると判定され、入力画像入れ替えフラグがONに設定される。入力画像入れ替えフラグがONに設定された画素は、出力されるとき、入力画像の画素と入れ替えられて(元の状態に戻されて)出力される。

## 【0286】

ステップS865において、画像合成部747は、全ての画素についてチェックしたか否かを判定し、まだ全画素をチェックしていないと判定された場合、ステップS861に戻り、それ以降の処理が繰り返し実行される。ステップS865において、全ての画素についてチェックしたと判定された場合、ステップS866において、画像合成部747は、画像が補正され、ぼけが除去された補正対象エリア52内の画像と、背景53の画像を合成し、画像ディスプレイ23に出力する。

## 【0287】

このようにして、画素毎に、積和演算結果を出力するか、または入力画像の画素をそのまま出力するかが判定される。このようにすることで、入力画像の中で、もともとぼけていない部分に対して、画像の修正を行うことにより、かえって画像が劣化してしまうことを防止することができる。

## 【0288】

この点について、図63と図64を参照して、さらに詳しく説明する。図63は、画像合成部747の構成例を示すブロック図である。積和演算部746からの出力結果が、ブロック切り出し部901に入力され、ブロック切り出し部901は、図64に示されるように、注目画素a5を中心とした9(=3×3)個の画素a1乃至a9を切り出し、分散演算部802に出力する。分散演算部802は、分散度を次のようにして演算する。

## 【0289】

## 【数15】

$$v = \sum_{* = 1}^9 (a^* - m)^2 \quad \dots (17)$$

## 【0290】

ここで、mは、ブロック内の9個の画素(のレベル)の平均値を表し、vは、それぞれの画素の平均値との差の2乗の総和であり、ブロック内の画素の分散度を表す。ステップS861においては、このようにして分散度が演算され、演算結果が閾値判定部903に出力される。

## 【0291】

閾値判定部903は、分散演算部902からの出力結果(分散度)と予め設定された閾値を比較し、分散度が閾値より大きいと判定された場合、画像合成部747は、注目画素に対応する入力画像入れ替えフラグをONに設定するように選択部904を制御する。分散度が閾値より大きくないと判定された場合、画像合成部747は、注目画素に対応する入力画像入れ替えフラグをOFFに設定するように選択部804を制御する。ステップS862乃至S864においては、このように、分散度が閾値より大きいかが判定され、判定結果に基づいて、入力画像入れ替えフラグが設定される。

## 【0292】

そして、切り替え部 905 により、選択部 904 による最終処理結果と、入力画像の画素が切り替えられ、出力される。すなわち補正対象エリア 52 内の画像の画素は、選択部 904 による最終処理結果とされ、背景 53 の画像の画素は、入力画像の画素となるように切り替えられる。

#### 【0293】

このようにして、オブジェクト 51 (図 3) が追尾され、オブジェクト 51 が含まれる補正対象エリア 52 内の画像のみが、画像のぼけが除去されるように修正(補正)され鮮明に表示される。一方、背景 53 の画像は、画像のぼけが除去されずに表示されるので、自然に、ユーザを、移動するオブジェクト 51 に注目させることができる。

#### 【0294】

以上においては、画像補正部 22 により、撮像部 21 により撮像された画像の中の補正対象エリア 52 内の画像について、画像のぼけが除去されるように、補正される例について説明したが、画像補正部 22 により、画像のぼけが除去されることなく、補正対象エリア 52 内の画像について、例えば、画像を構成する各画素の輝度や色などの設定が変更され、単純にハイライト表示されるように、画像が補正されるようにしてもよい。このようにすることで、ユーザがオブジェクト 51 を正確に視認することができなくなるおそれはあるものの、やはり、自然に、ユーザを、移動するオブジェクト 51 に注目させることができ、また、画像のぼけが除去されるように補正される場合と比較して、画像補正部 22 を、より簡単な構成とすることができ、その結果、監視カメラシステム 1 を、より低コストで実現することができる。

#### 【0295】

なお、上述した一連の処理をハードウェアで実現するか、ソフトウェアで実現するかは問わない。上述した一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、汎用のパーソナルコンピュータなどに、ネットワークやリムーバブルメディアなどの記録媒体からインストールされる。

#### 【0296】

また、本明細書において上述した一連の処理を実行するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

#### 【図面の簡単な説明】

#### 【0297】

- 【図 1】 本発明を適用した監視カメラシステムの構成例を示すブロック図である。
- 【図 2】 監視処理を説明するフローチャートである。
- 【図 3】 図 1 の監視カメラシステムにより表示される画像の例を示す図である。
- 【図 4】 図 3 の補正対象エリアの移動の例を示す図である。
- 【図 5】 図 1 のオブジェクト追尾部の構成例を示すブロック図である。
- 【図 6】 追尾処理を説明するフローチャート図である。
- 【図 7】 追尾対象が回転する場合の追尾を説明する図である。
- 【図 8】 オクルージョンが起きる場合の追尾を説明する図である。
- 【図 9】 シーンチェンジが起きる場合の追尾を説明する図である。
- 【図 10】 通常処理を説明するフローチャートである。
- 【図 11】 通常処理の初期化処理を説明するフローチャートである。
- 【図 12】 乗り換え候補抽出処理を説明する図である。
- 【図 13】 領域推定関連処理部の構成例を示すブロック図である。
- 【図 14】 領域推定関連処理を説明するフローチャートである。
- 【図 15】 領域推定処理を説明するフローチャートである。
- 【図 16】 サンプル点を決定する処理を説明する図である。
- 【図 17】 サンプル点を決定する処理を説明する図である。

- 【図18】 サンプル点を決定する処理を説明する図である。
- 【図19】 サンプル点を決定する処理を説明する図である。
- 【図20】 領域推定範囲の更新処理を説明するフローチャートである。
- 【図21】 領域推定範囲の更新を説明する図である。
- 【図22】 領域推定範囲の更新を説明する図である。
- 【図23】 領域推定範囲の更新処理の他の例を説明するフローチャートである。
- 【図24】 領域推定範囲の更新を説明するフローチャートである。
- 【図25】 乗り換え候補抽出処理を説明するフローチャートである。
- 【図26】 テンプレート作成処理を説明するフローチャートである。
- 【図27】 テンプレート作成を説明する図である。
- 【図28】 テンプレート作成を説明する図である。
- 【図29】 テンプレートと追尾点の位置関係を説明する図である。
- 【図30】 領域推定関連処理部の他の構成例を示すブロック図である。
- 【図31】 領域推定処理の他の例を説明するフローチャートである。
- 【図32】 同色領域の成長を説明する図である。
- 【図33】 追尾点の同色領域と領域推定結果を説明する図である。
- 【図34】 乗り換え候補抽出処理の他の例を説明するフローチャートである。
- 【図35】 例外処理を説明するフローチャートである。
- 【図36】 例外処理の初期化処理を説明するフローチャートである。
- 【図37】 テンプレートの選択を説明する図である。
- 【図38】 探索範囲の設定を説明する図である。
- 【図39】 継続判定処理を説明するフローチャートである。
- 【図40】 動き推定部の構成例を示すブロック図である。
- 【図41】 動き推定処理を説明するフローチャートである。
- 【図42】 アクティビティの算出を説明する図である。
- 【図43】 評価値とアクティビティの関係を説明する図である。
- 【図44】 統合処理を説明するフローチャートである。
- 【図45】 背景動き推定部の構成例を示すブロック図である。
- 【図46】 背景動き推定処理を説明するフローチャートである。
- 【図47】 シーンチェンジ検出部53の構成例を示すブロック図である。
- 【図48】 シーンチェンジ検出処理を説明するフローチャートである。
- 【図49】 図1の画像補正部の構成例を示すブロック図である。
- 【図50】 画像補正部の制御信号の例を示す図である。
- 【図51】 画像のぼけの原理を説明する図である。
- 【図52】 画像のぼけの原理を説明する図である。
- 【図53】 画像のぼけの原理を説明する図である。
- 【図54】 画像のぼけの原理を説明する図である。
- 【図55】 パラメータコードの組み合わせの例を示す図である。
- 【図56】 画像のエッジ部分を説明する図である。
- 【図57】 ぼけ修正処理を説明するフローチャートである。
- 【図58】 画像補正処理を説明するフローチャートである。
- 【図59】 画像特徴検出処理を説明するフローチャートである。
- 【図60】 画像特徴検出部の構成例を示すブロック図である。
- 【図61】 ブロック切り出し部により抽出される画像のブロックを説明する図である。
- 【図62】 画像合成処理を説明するフローチャートである。
- 【図63】 画像合成部の構成例を示すブロック図である。
- 【図64】 分散演算を説明する図である。

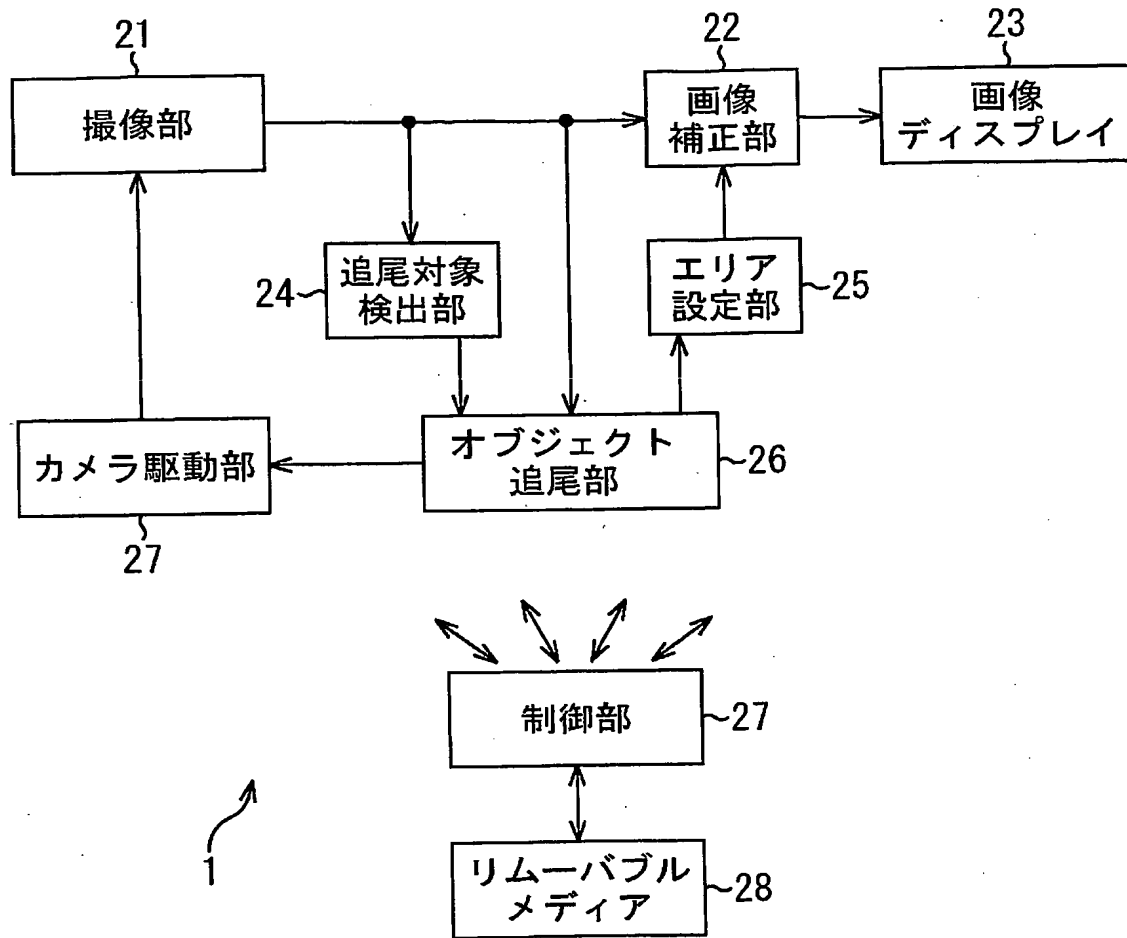
【符号の説明】

【0298】

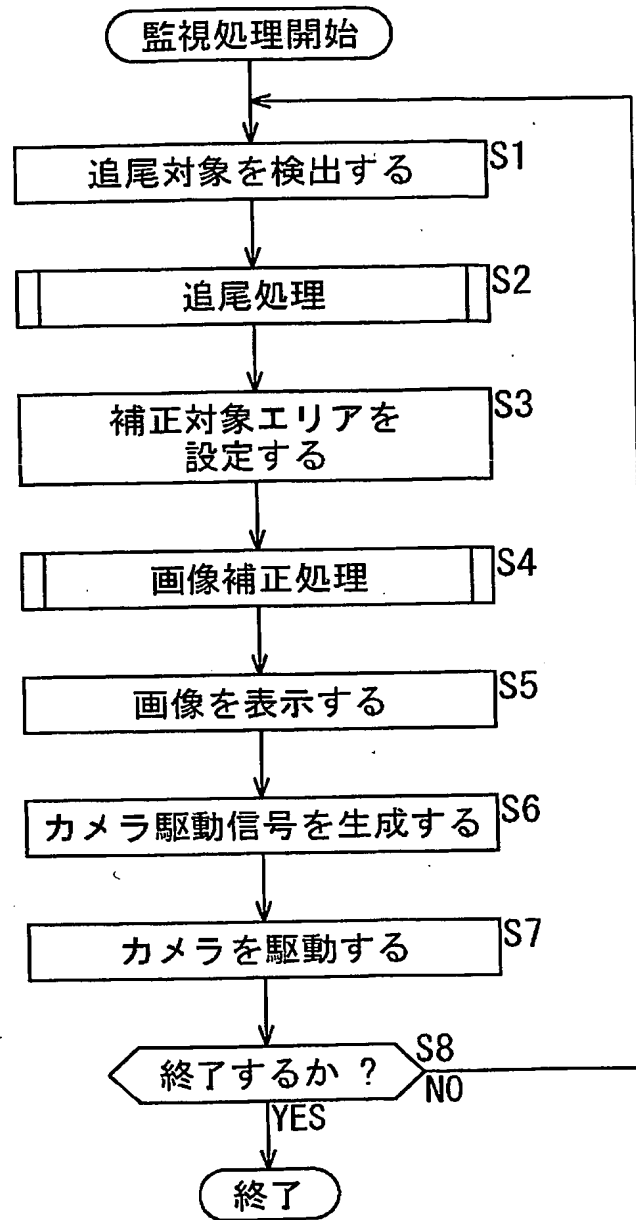
1 監視カメラシステム, 21 撮像部, 22 画像補正部, 24 追尾対象検出部, 25 エリア設定部, 26 オブジェクト追尾部, 51 テンプレートマッチング部, 52 動き推定部, 53 シーンチェンジ検出部, 54 背景動き推定部, 55 領域推定関連処理部, 56 乗り換え候補保持部, 57 追尾点決定部, 58 テンプレート保持部, 741 制御信号生成部, 742 画像特徴検出部, 743 アドレス演算部, 744 係数ROM, 745 領域抽出部, 746 積和演算部, 747 画像合成部

【書類名】 図面  
【図1】

図1

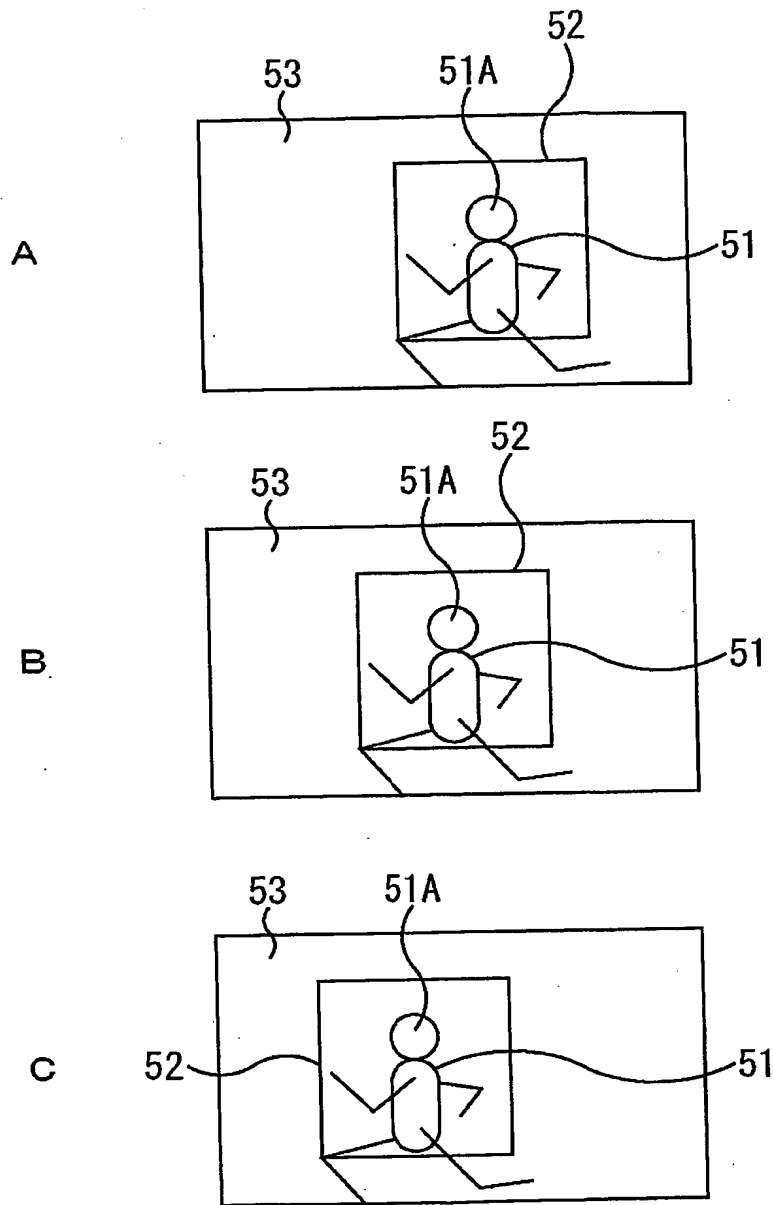


【図 2】  
図2



【図 3】

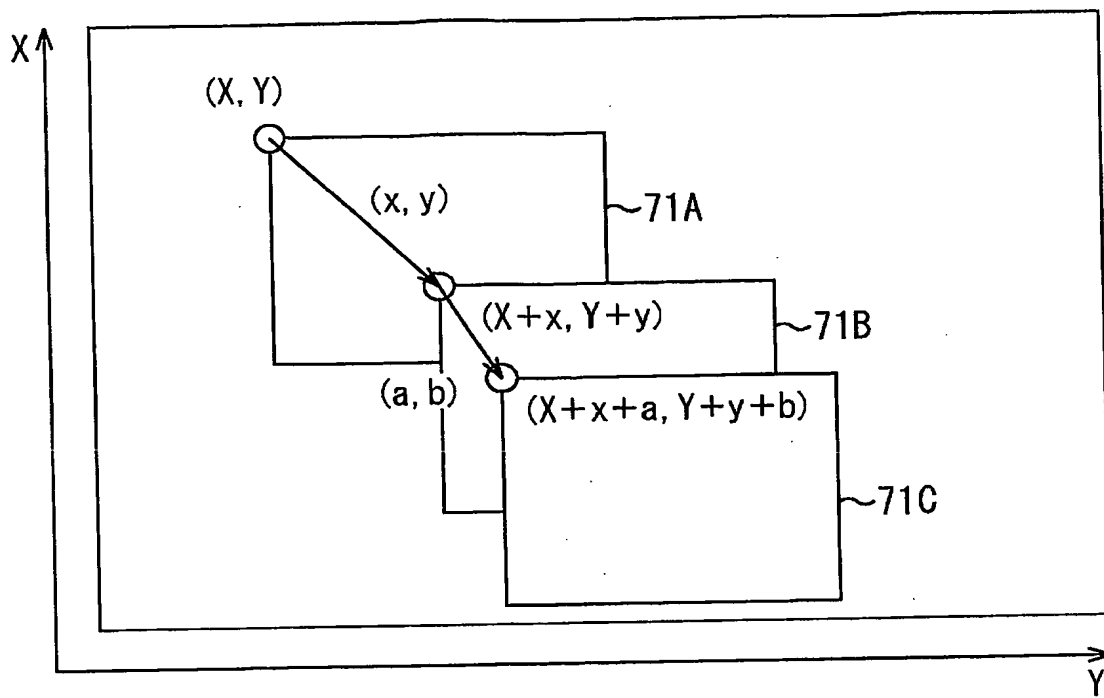
図3



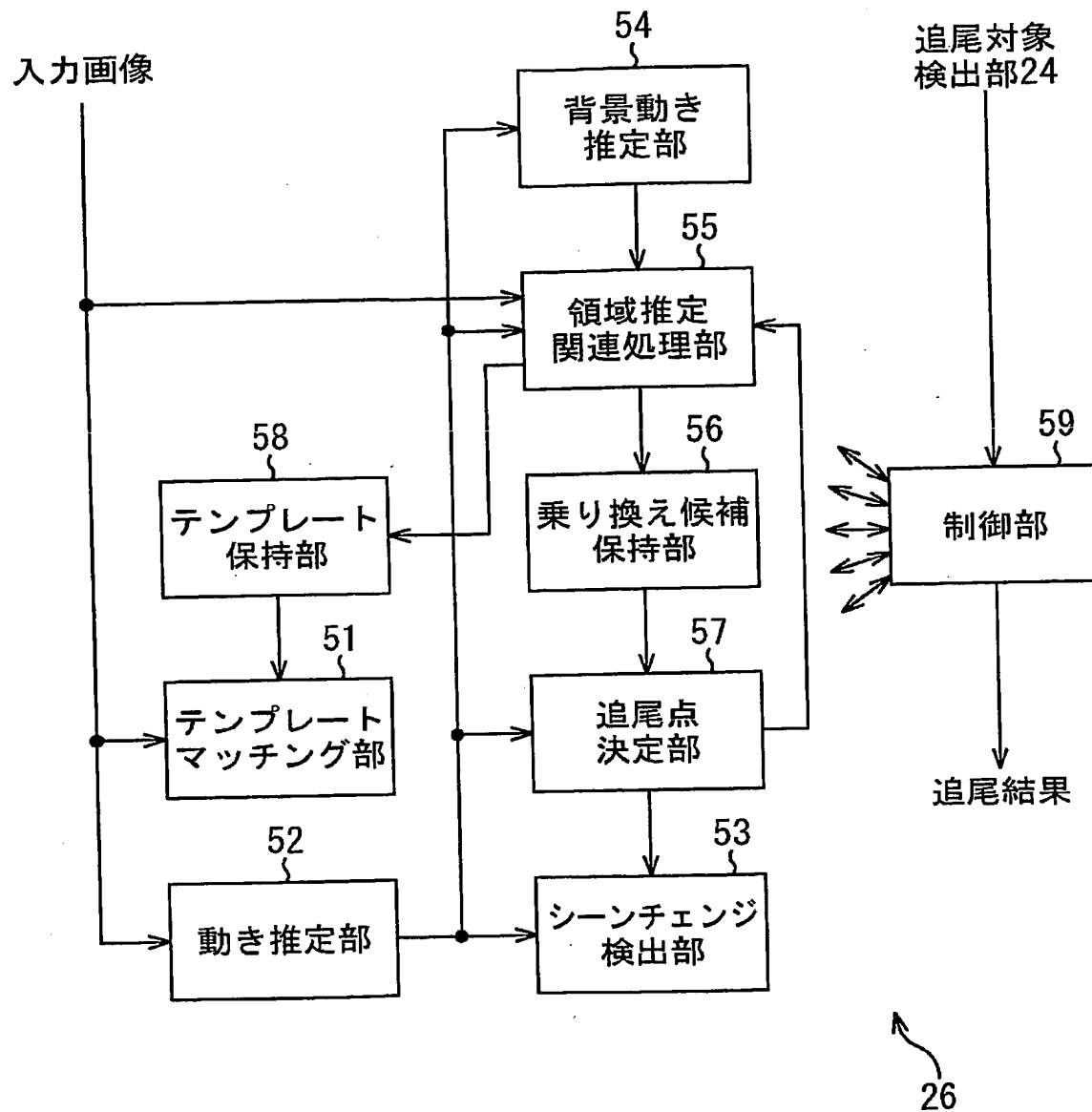


【図4】

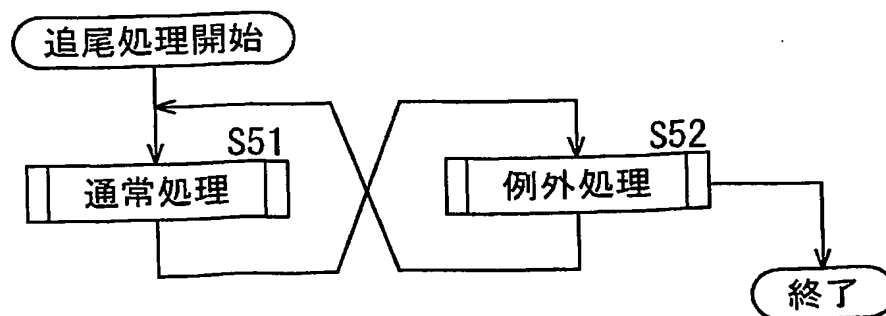
図4



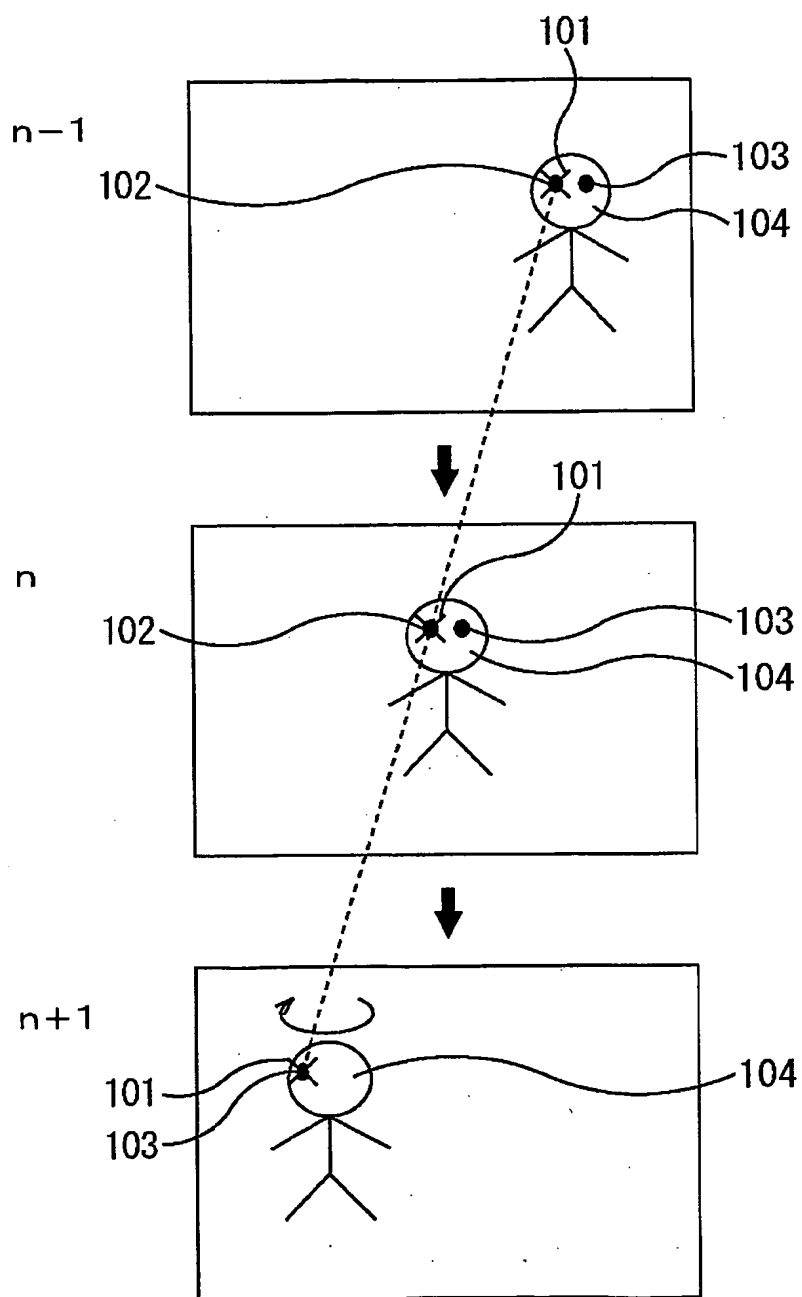
【図 5】  
図5



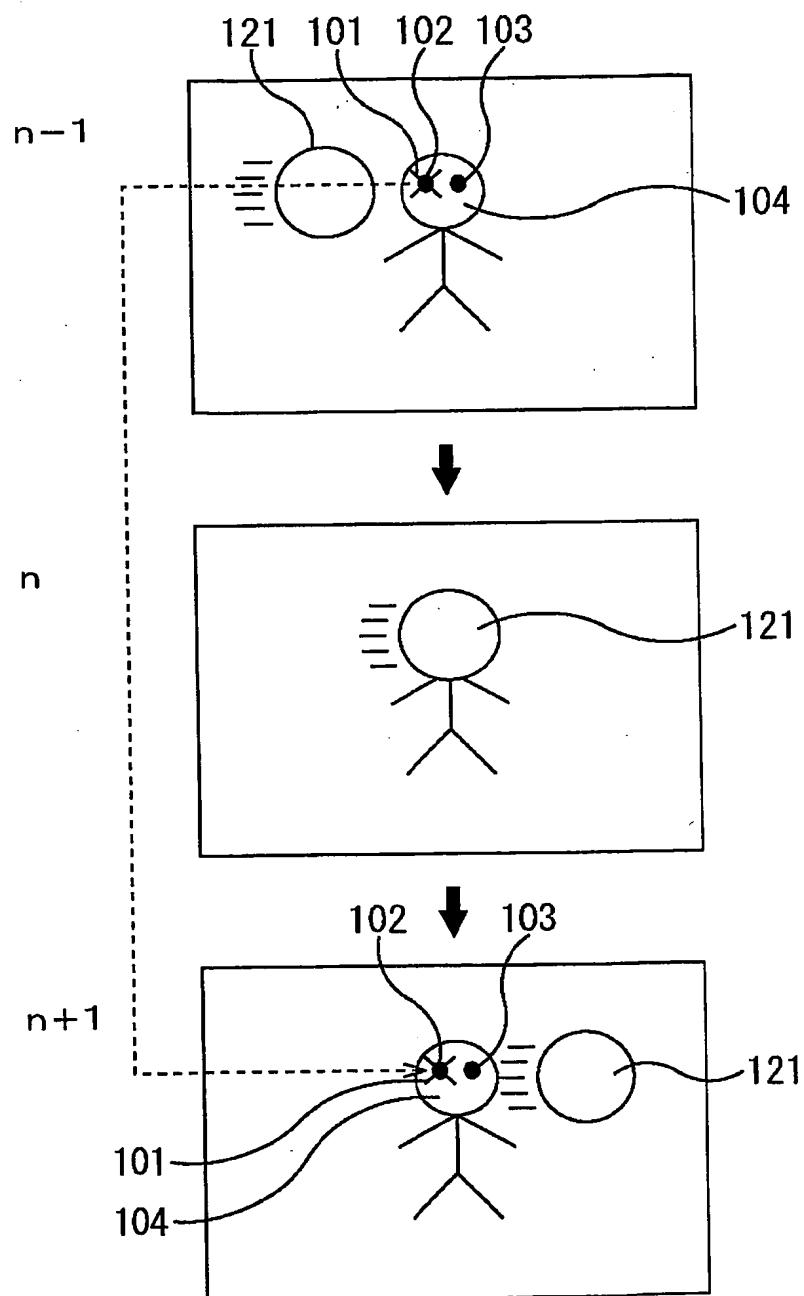
【図 6】  
図6



【図 7】  
図 7



【図 8】  
図8



【図 9】  
図9

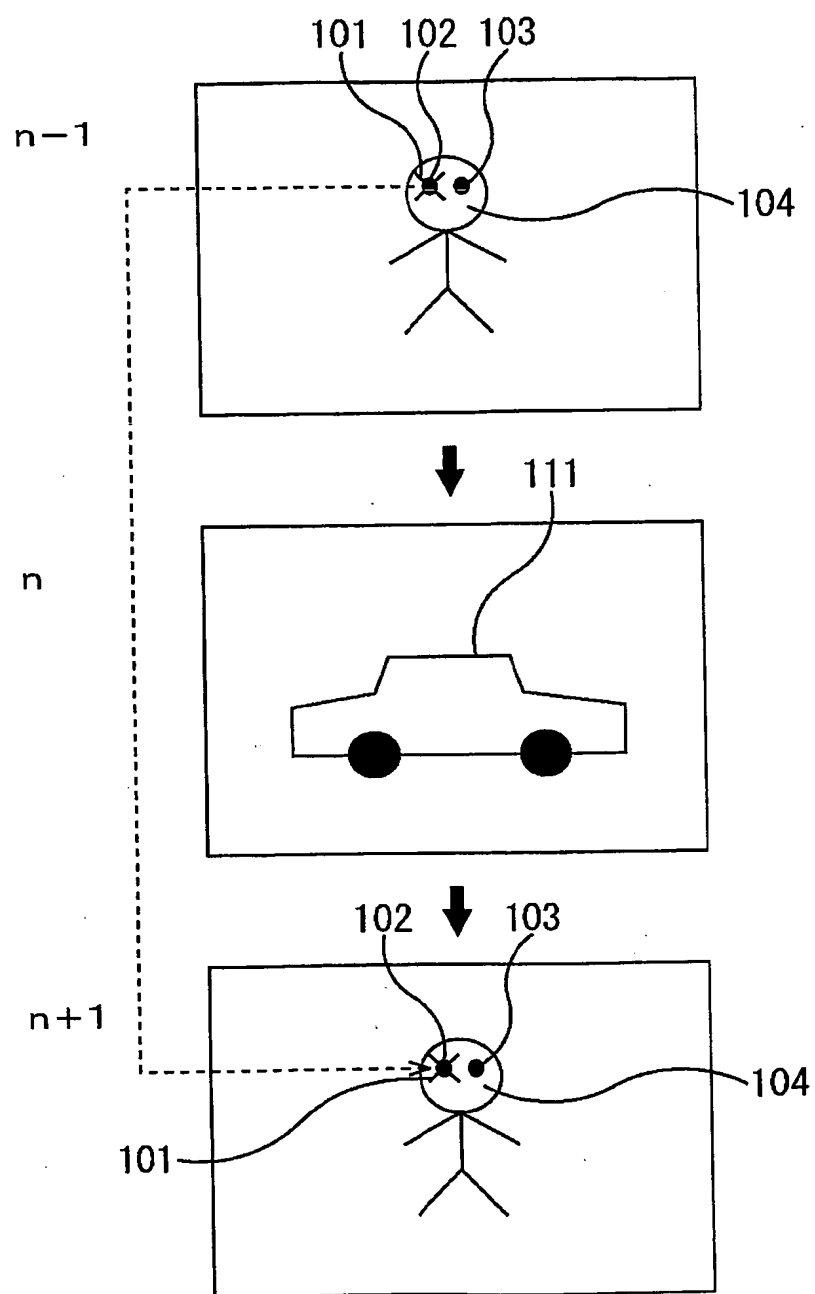
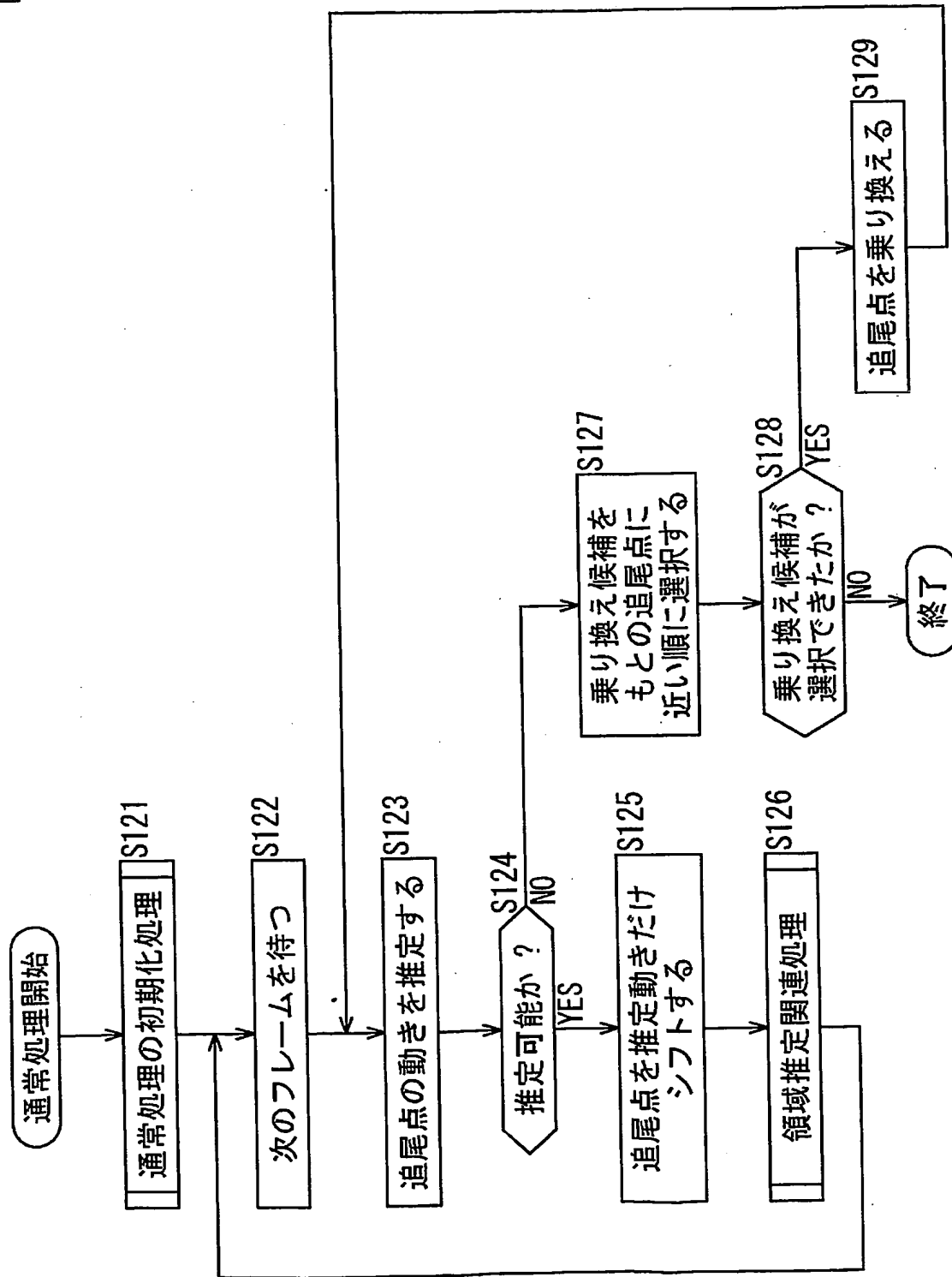
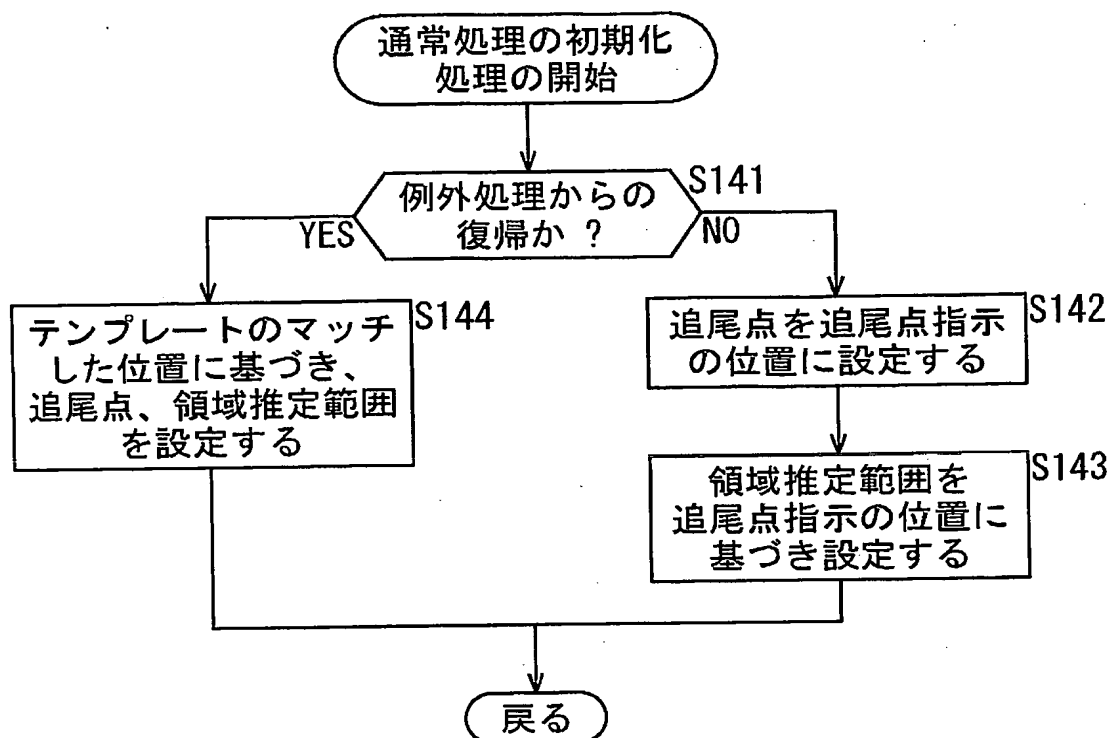


図10  
【図10】

【図11】  
図11

【図 12】  
図 12

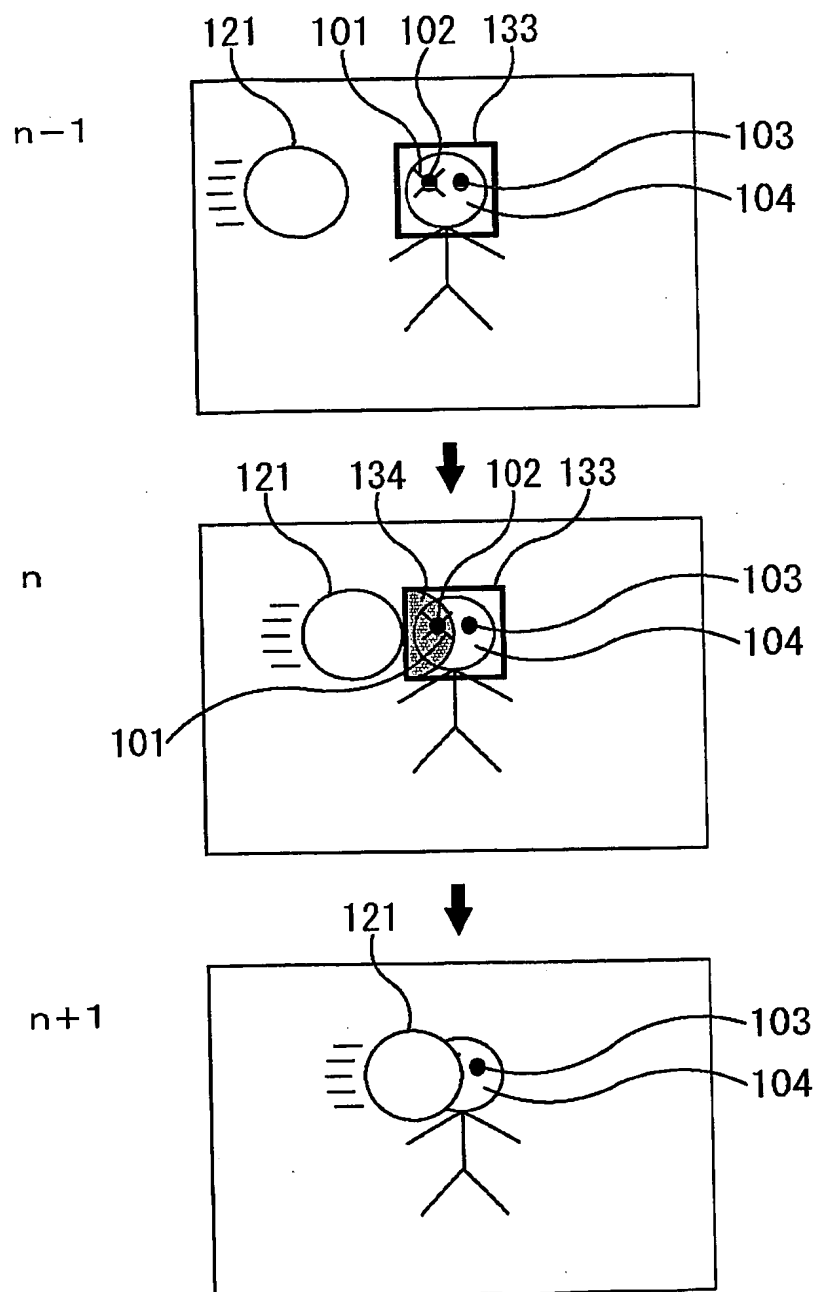
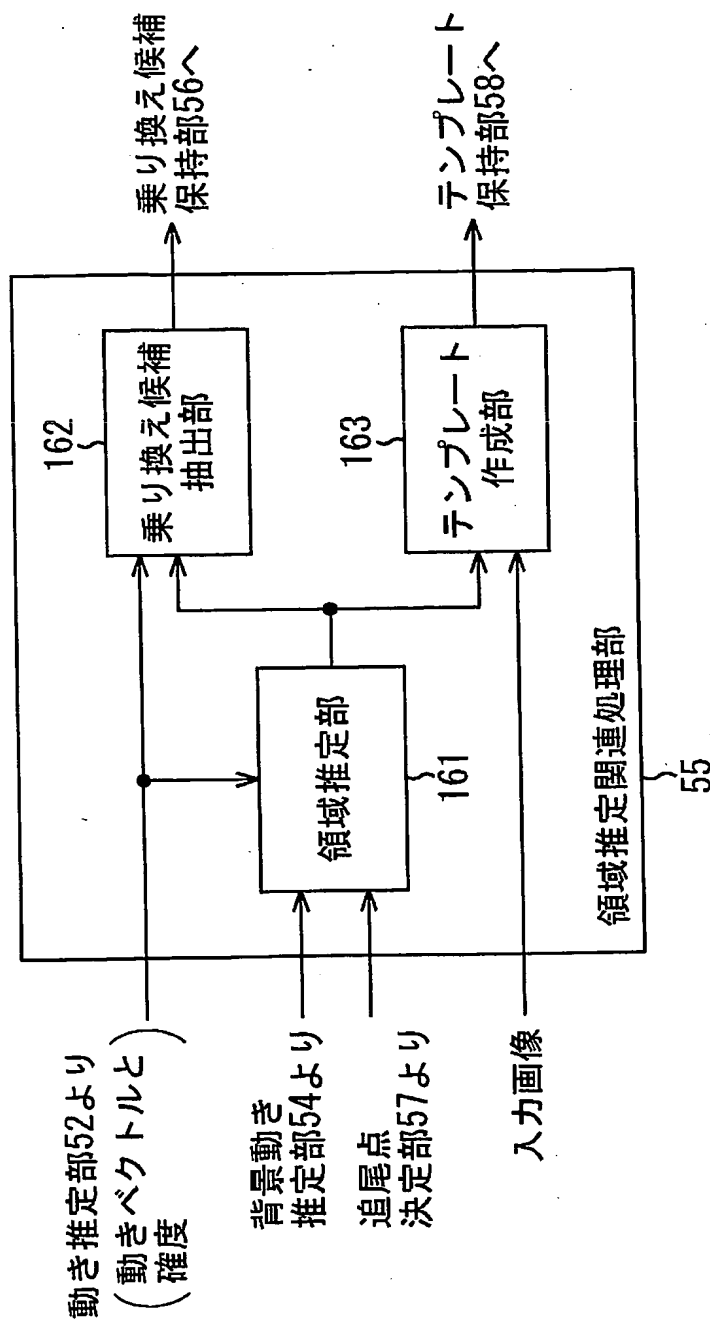


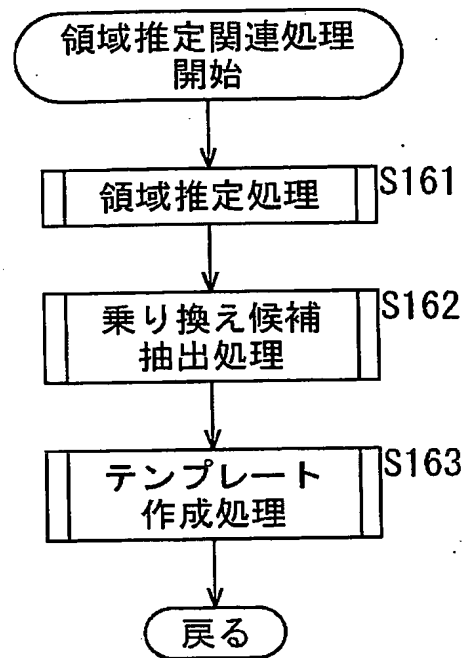


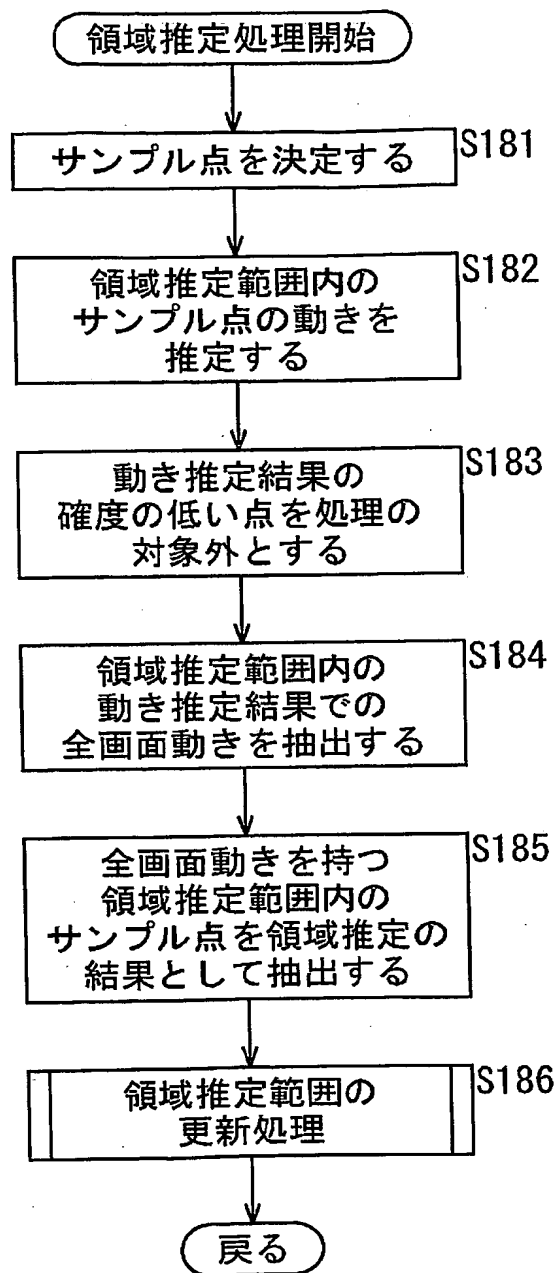
図13

【図13】



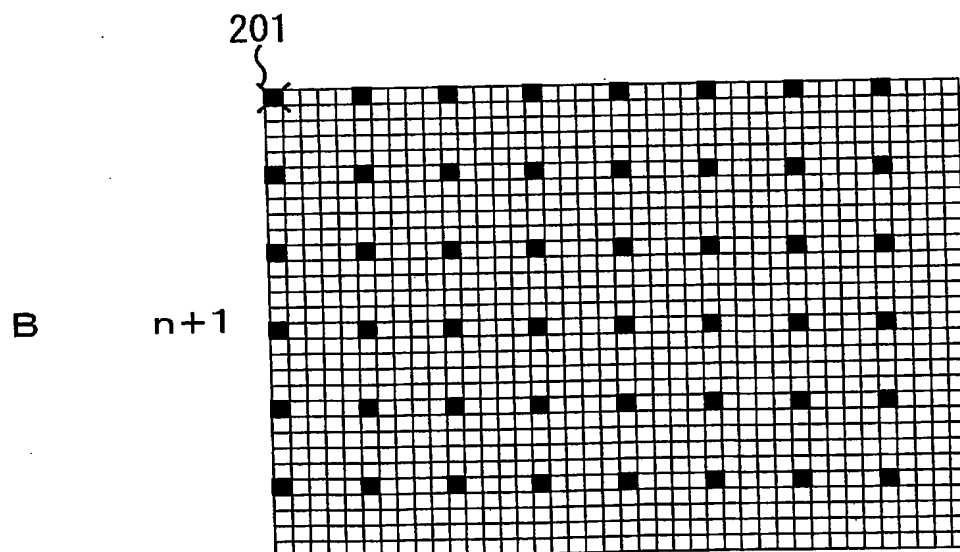
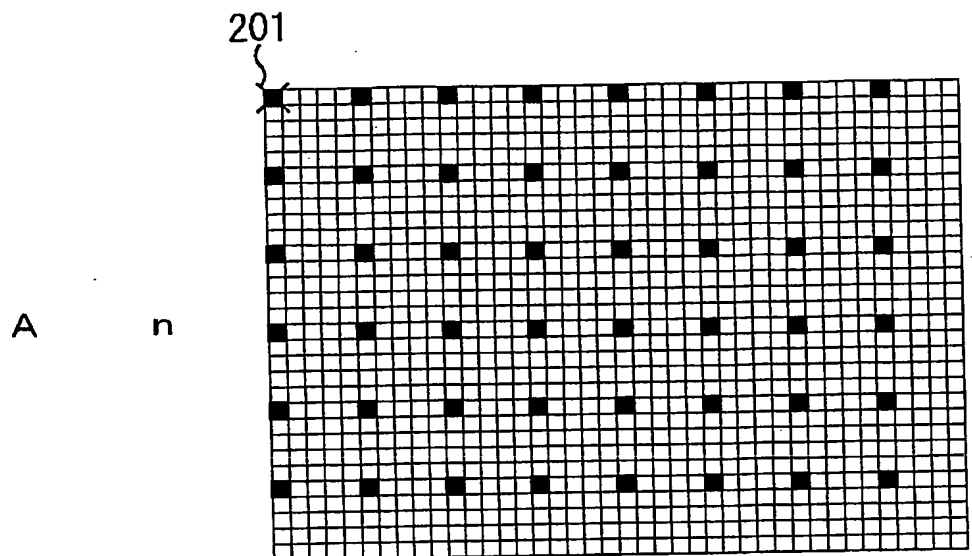
【図14】  
図14



【図15】  
図15

【図 16】

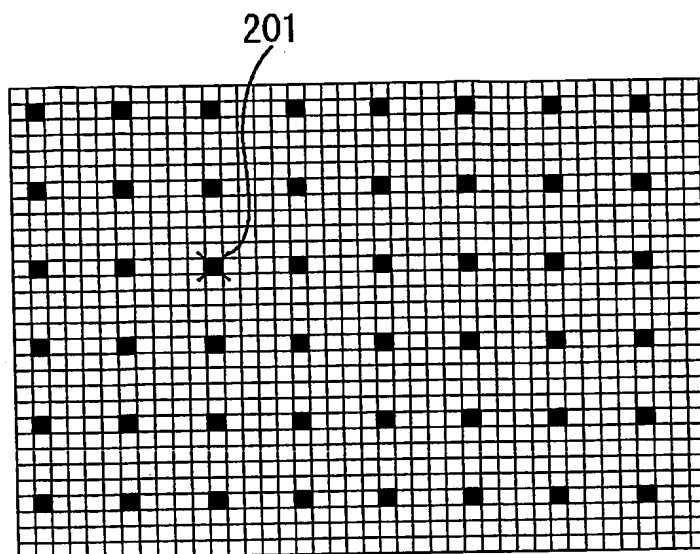
図16



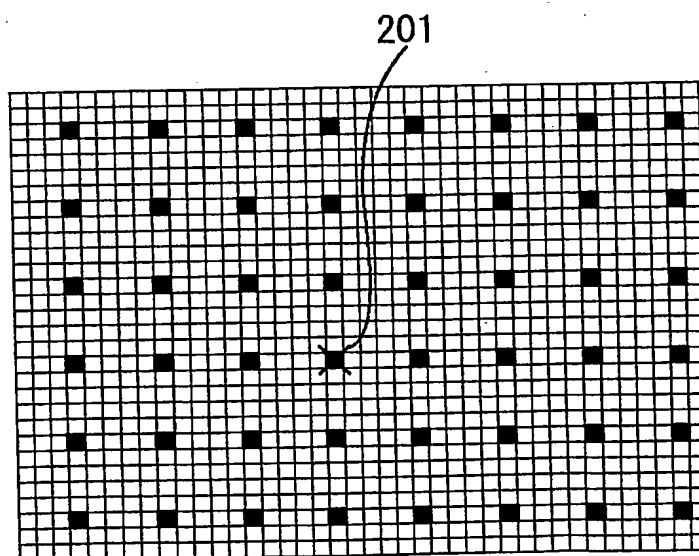
【図 17】

図 17

A      n



B      n+1

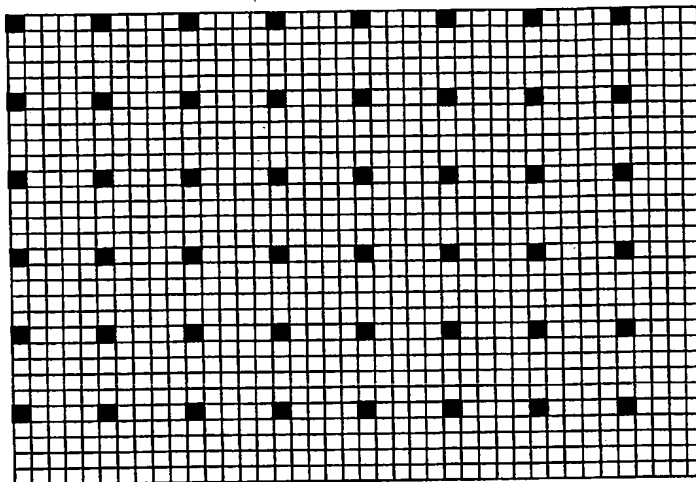


【図18】

図18

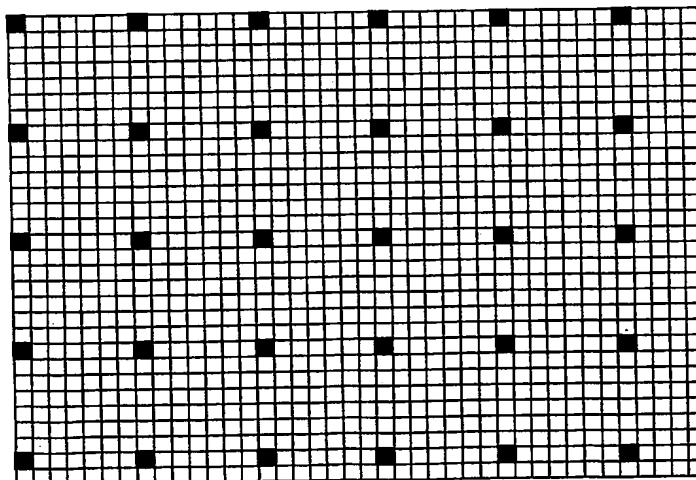
A

n



B

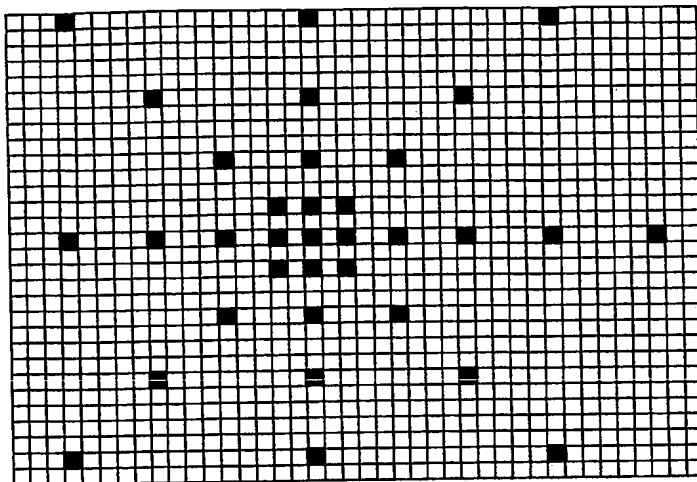
n+1



【図 19】

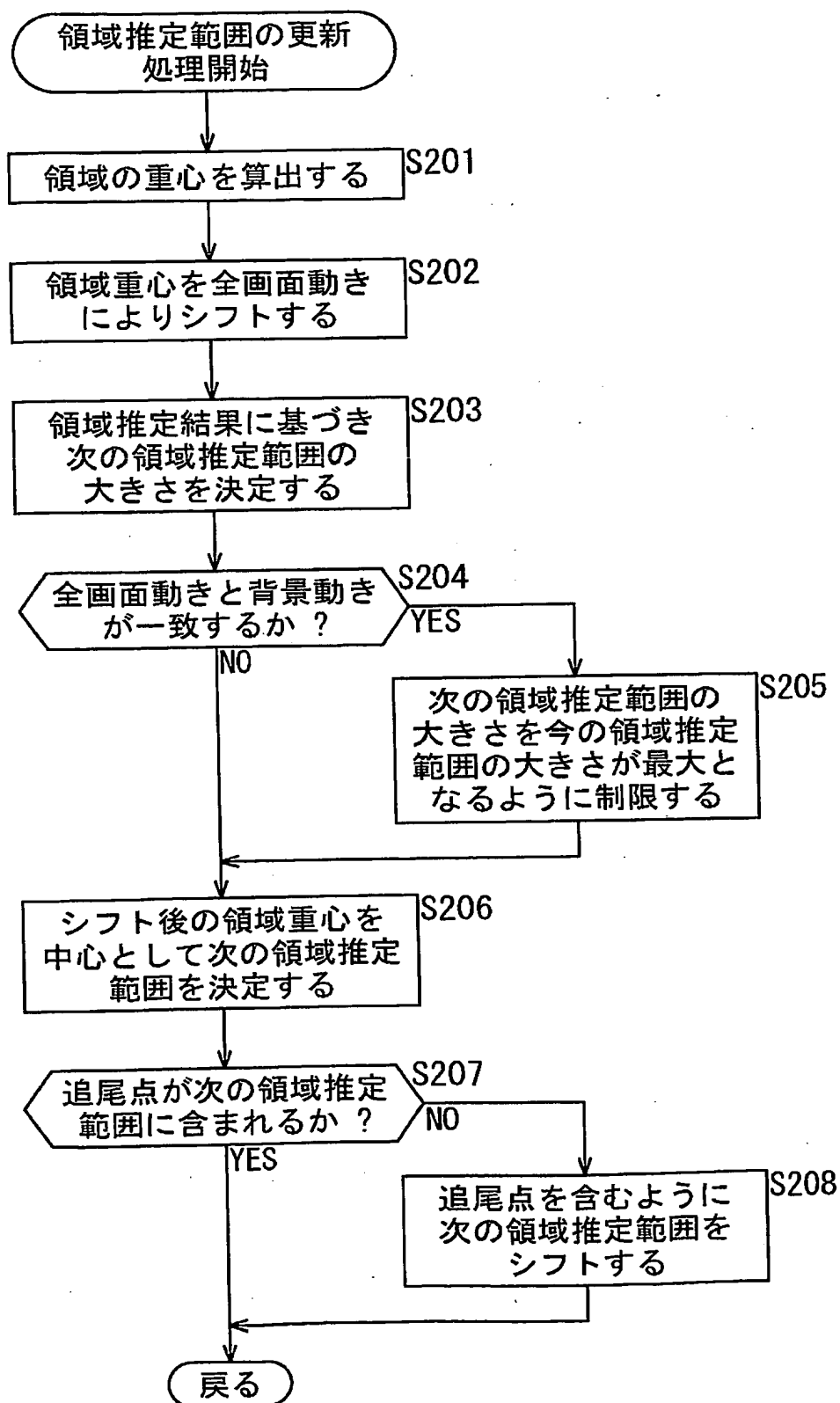
図19

n



【図 20】

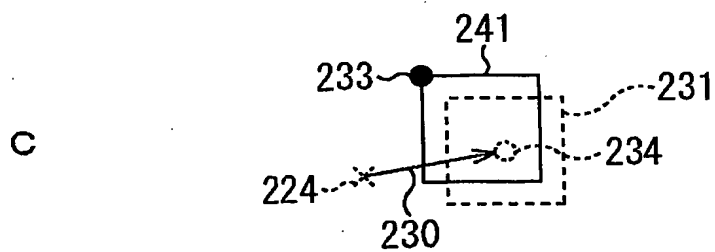
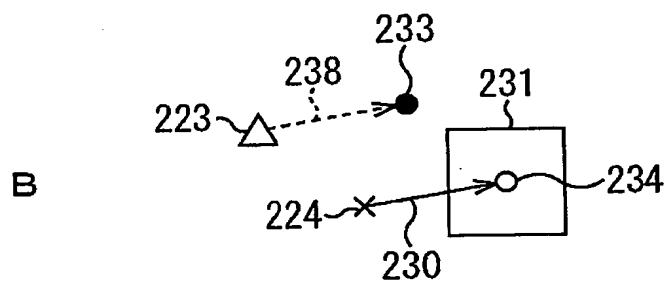
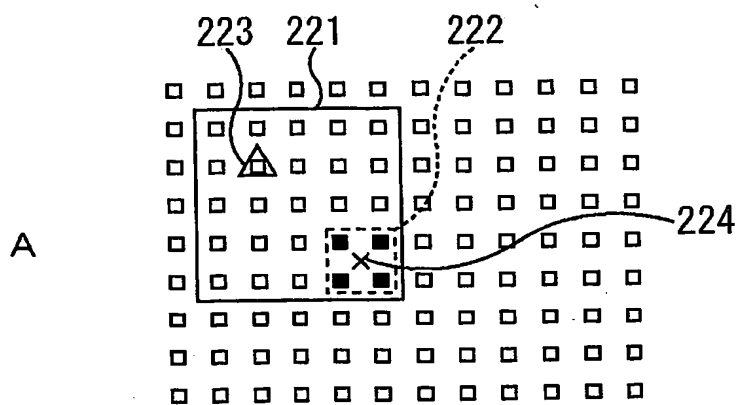
図20



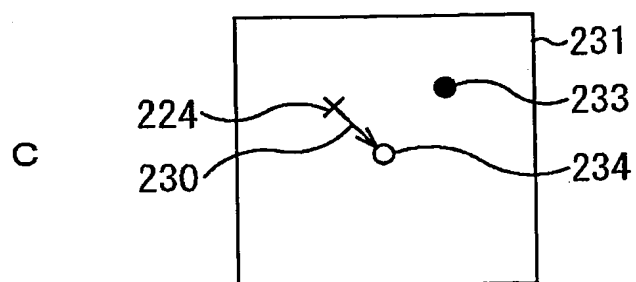
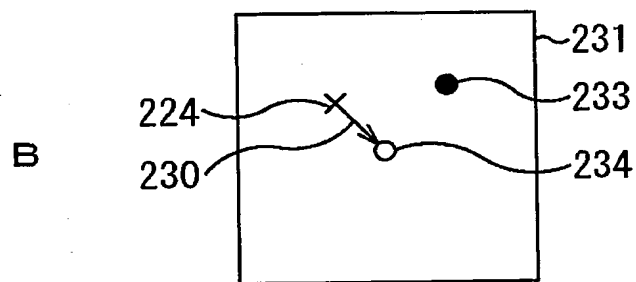
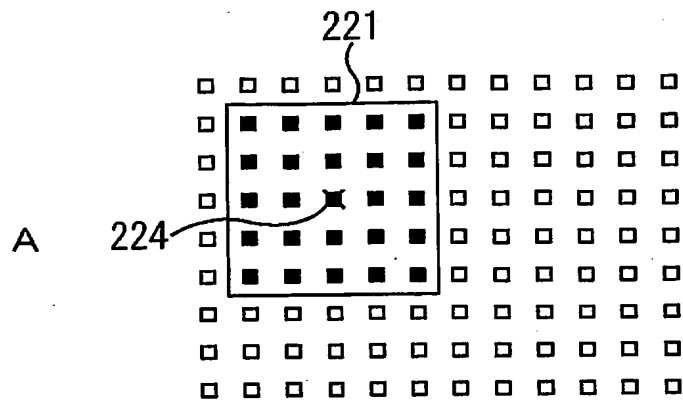


【図 21】

図21

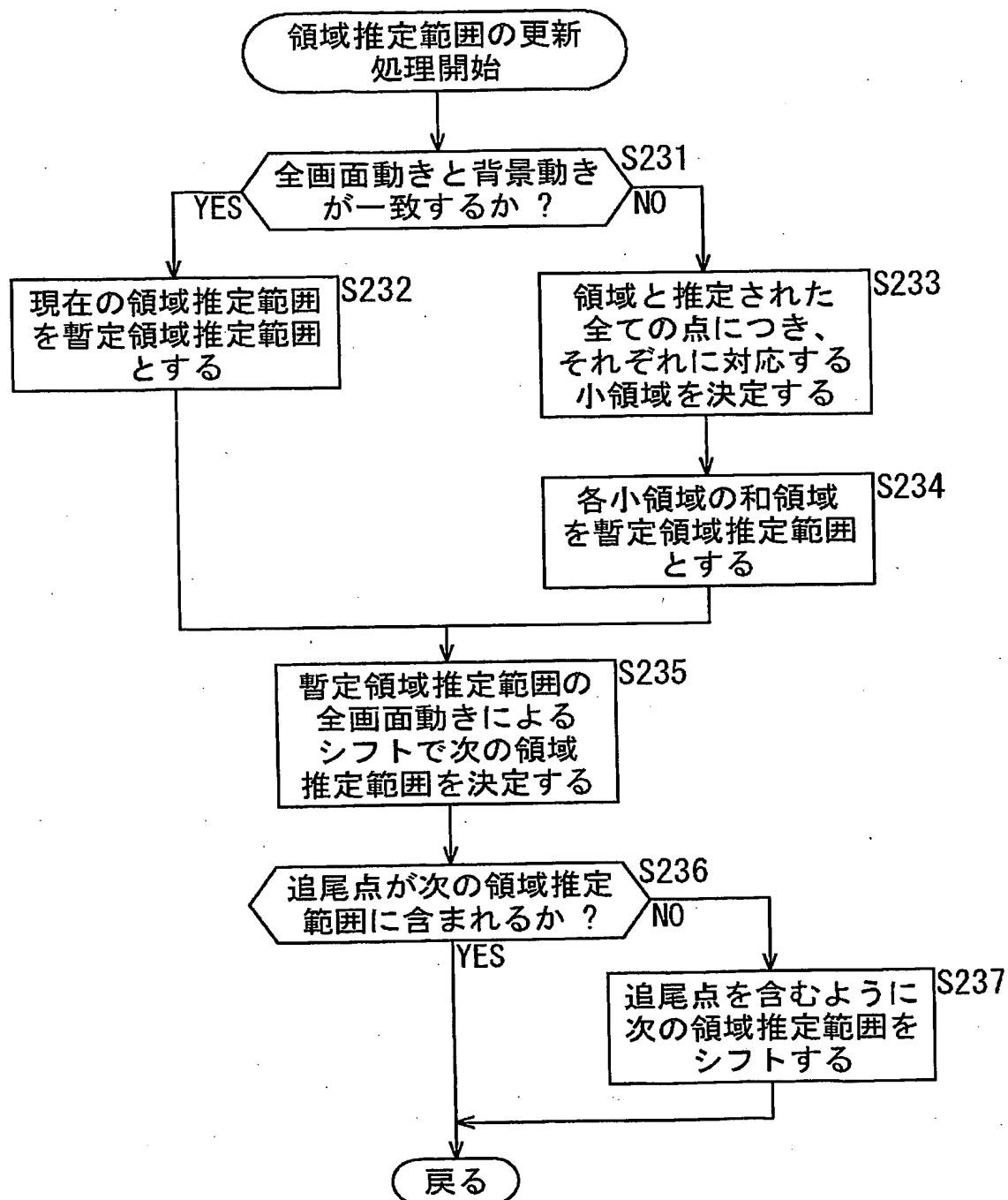


【図 22】  
図 22



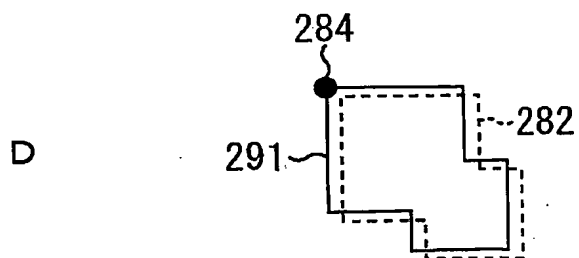
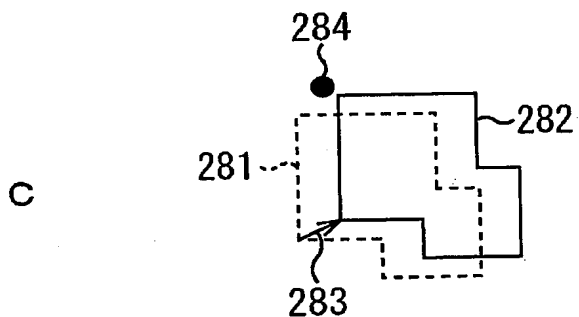
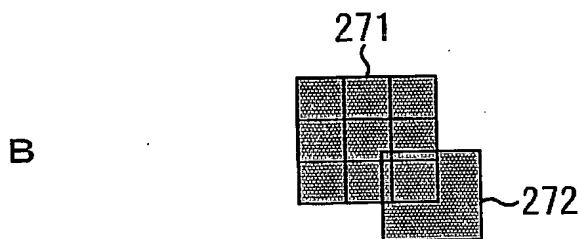
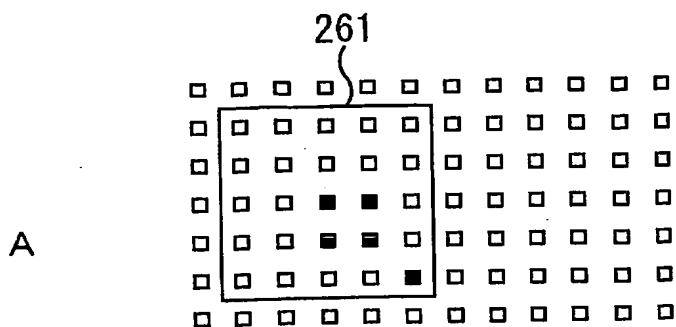
【図 23】

図23



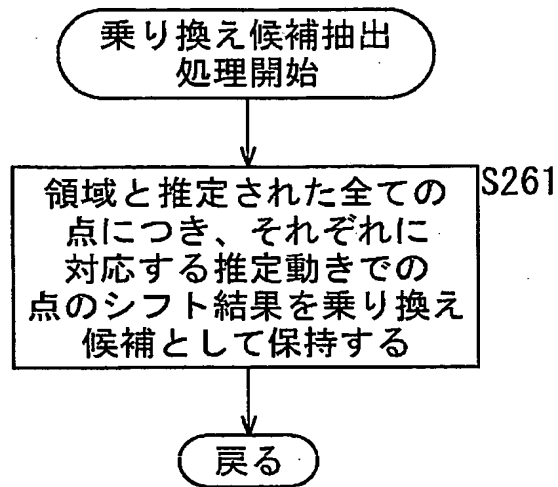
【図 24】

図24



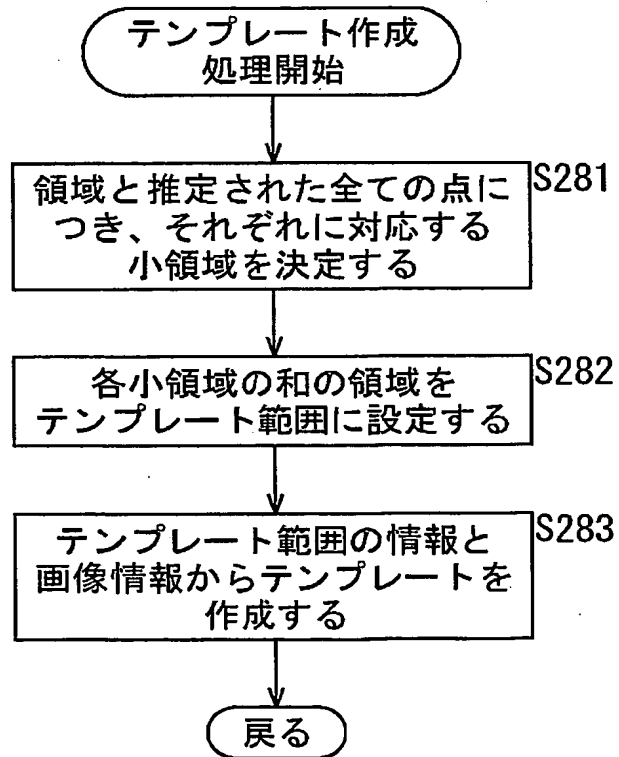
【図 25】

図25

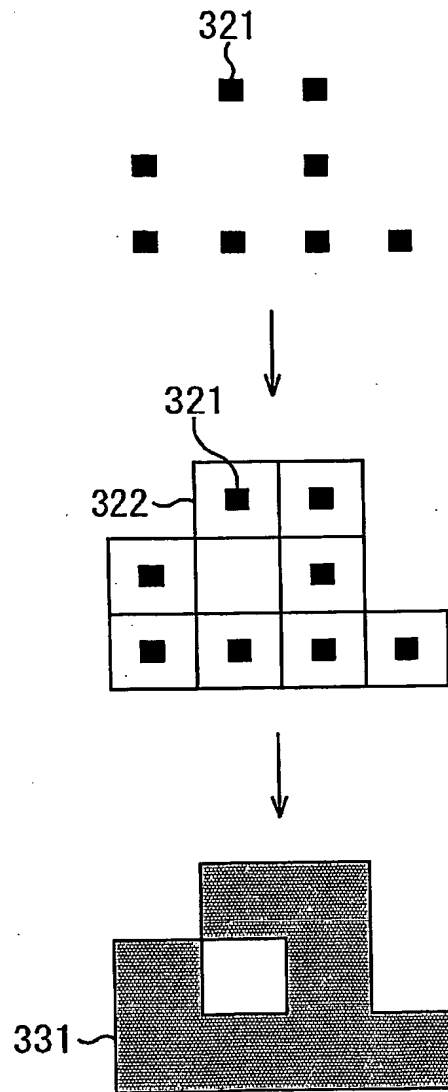


【図 26】

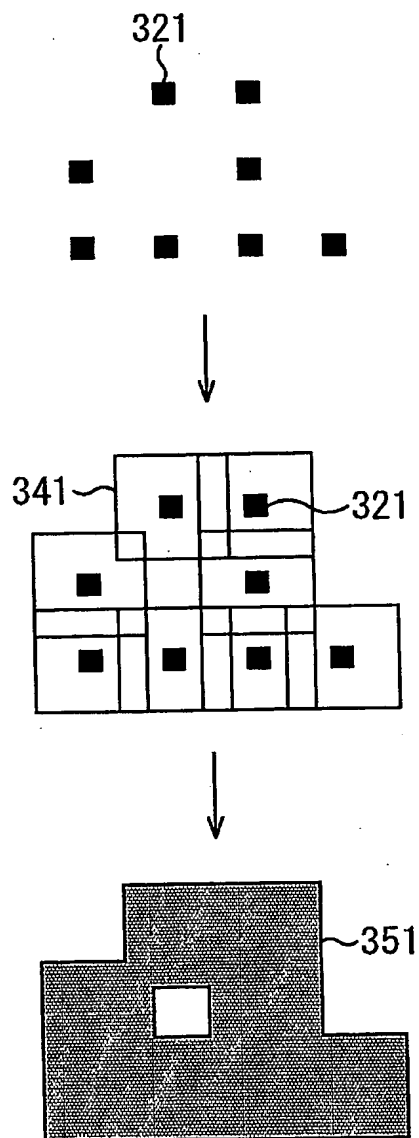
図26



【図 27】  
図 27



【図 28】  
図28



【図 29】  
図29

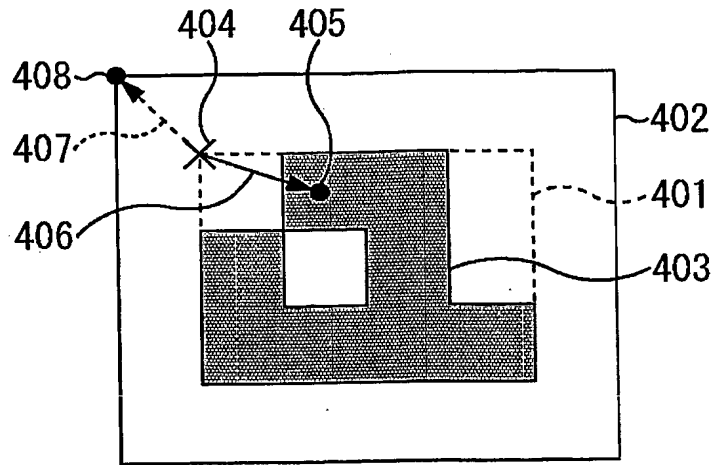
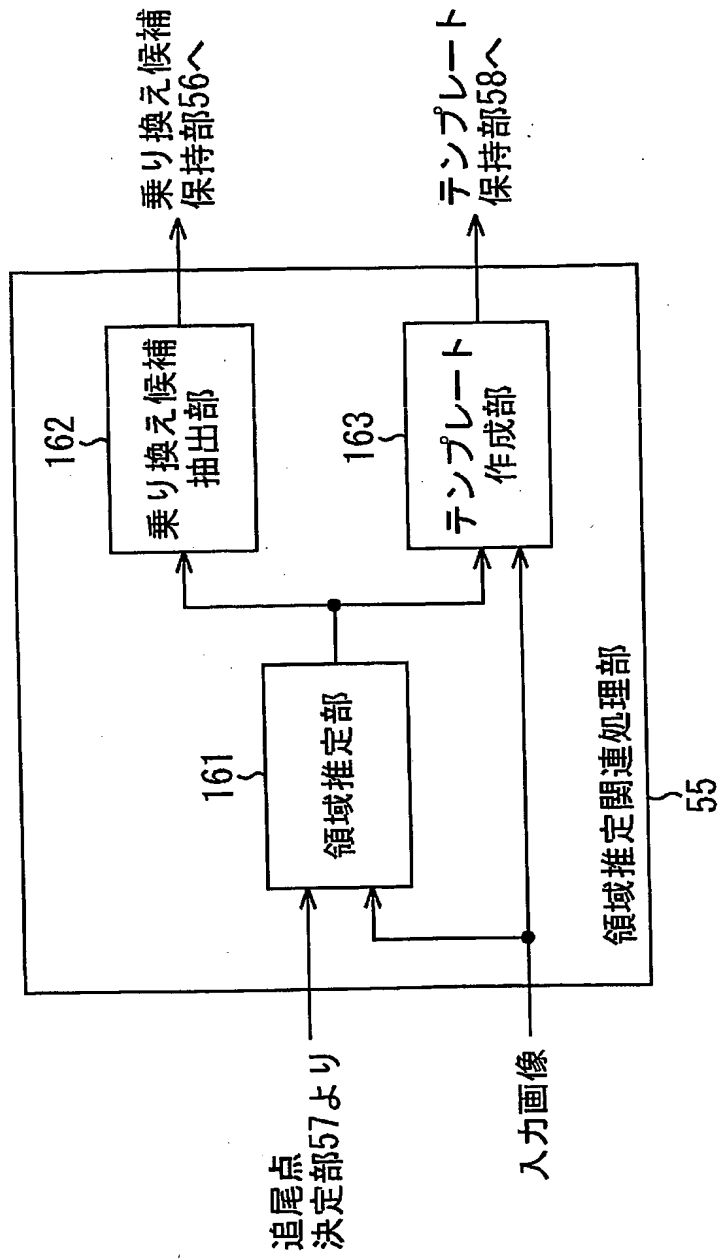


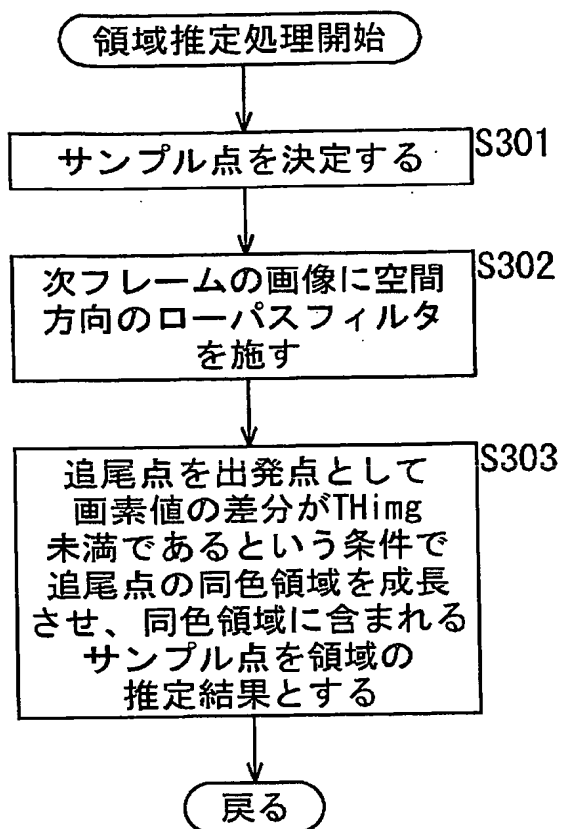


図30

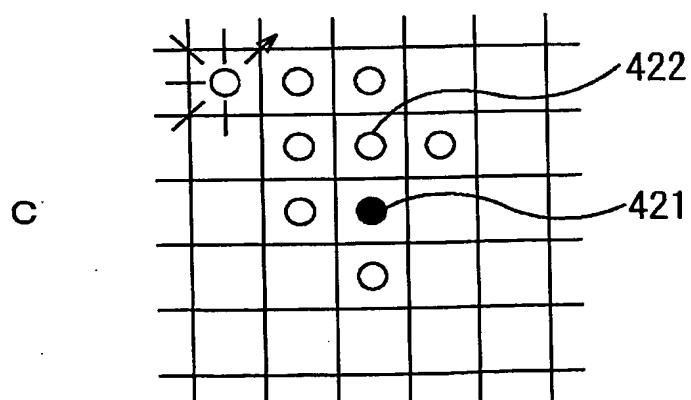
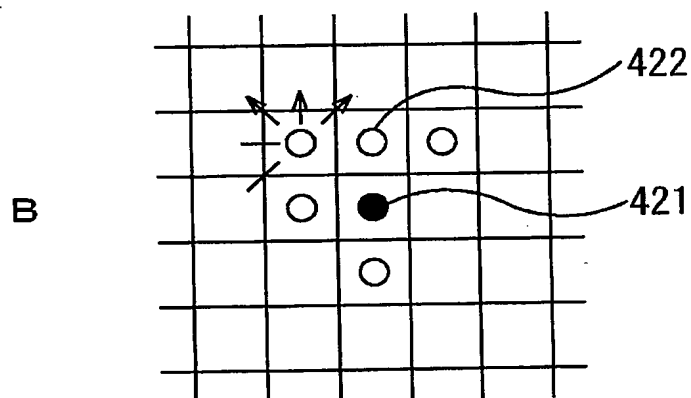
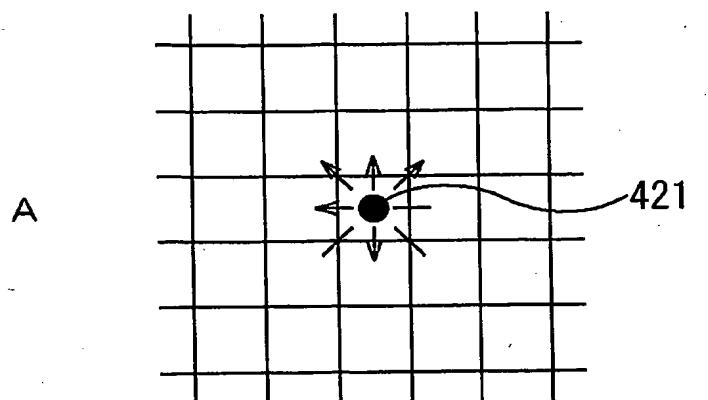
【図30】



【図 31】  
図31

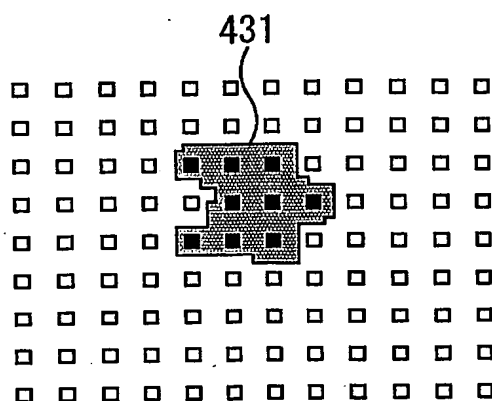


【図 32】  
図32



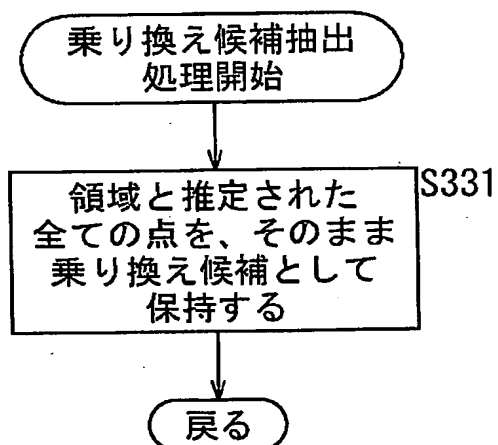
【図 33】

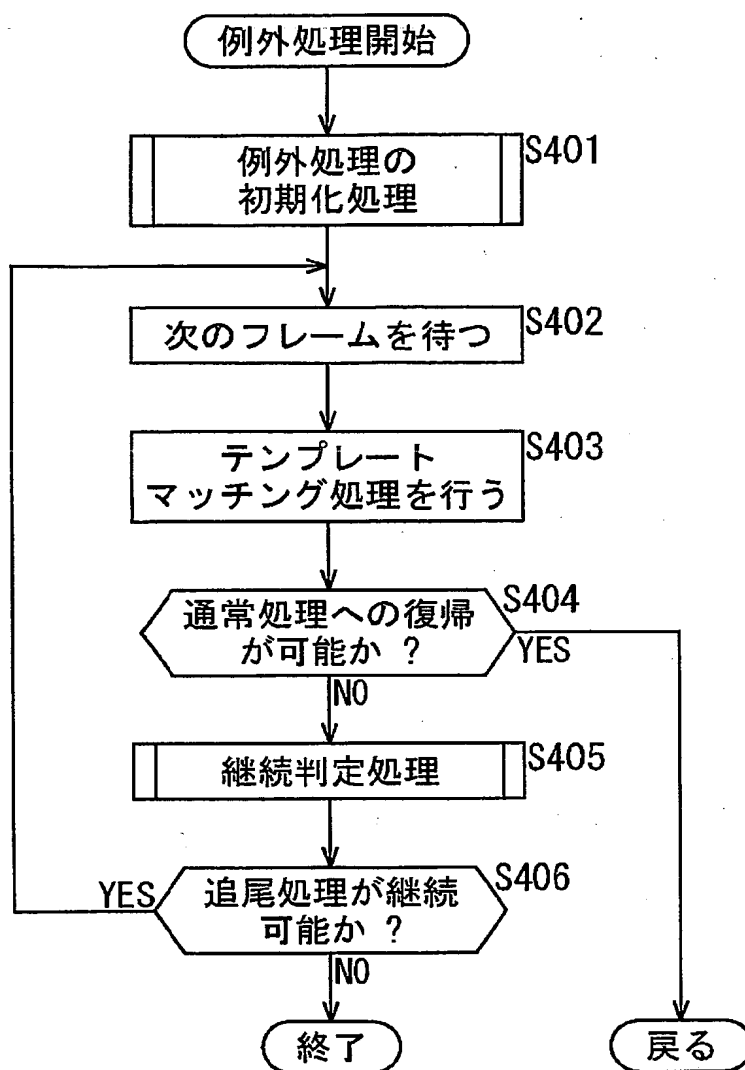
図33



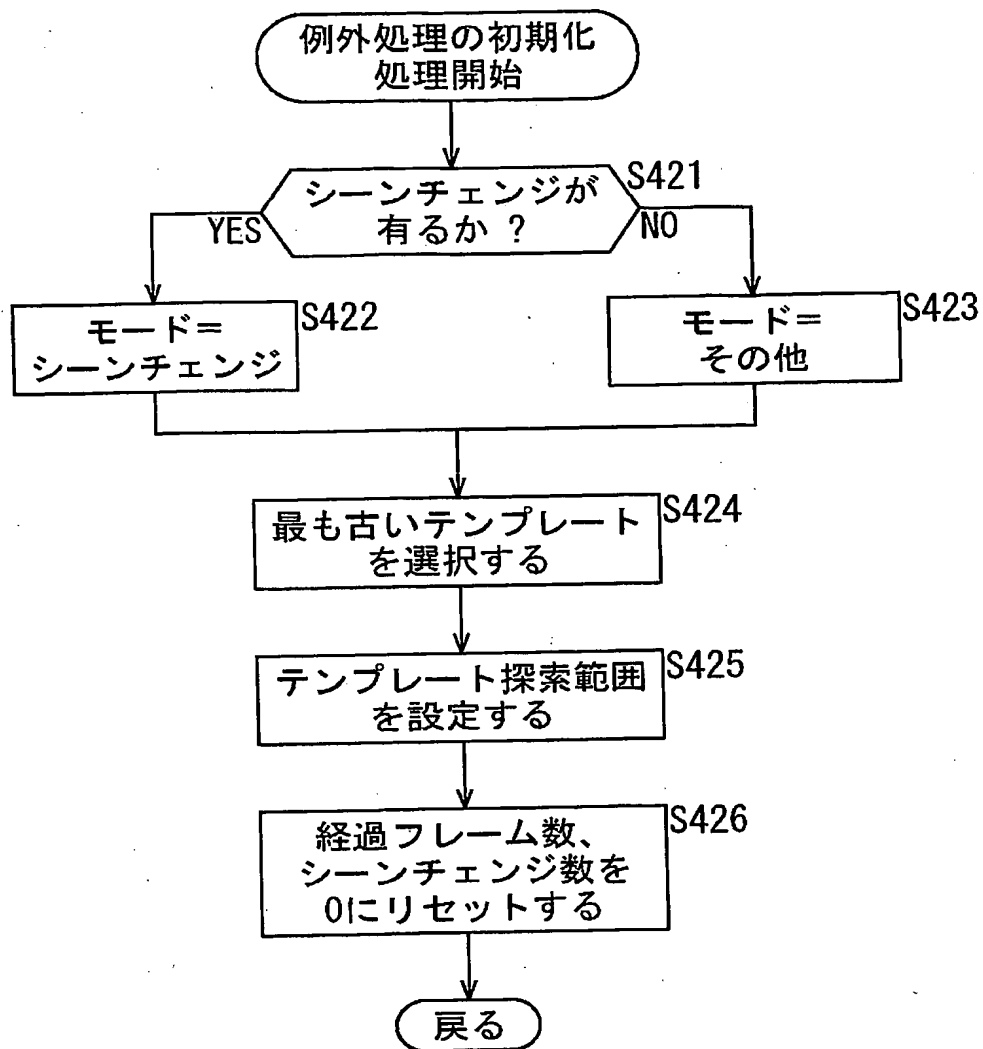
【図 34】

図34

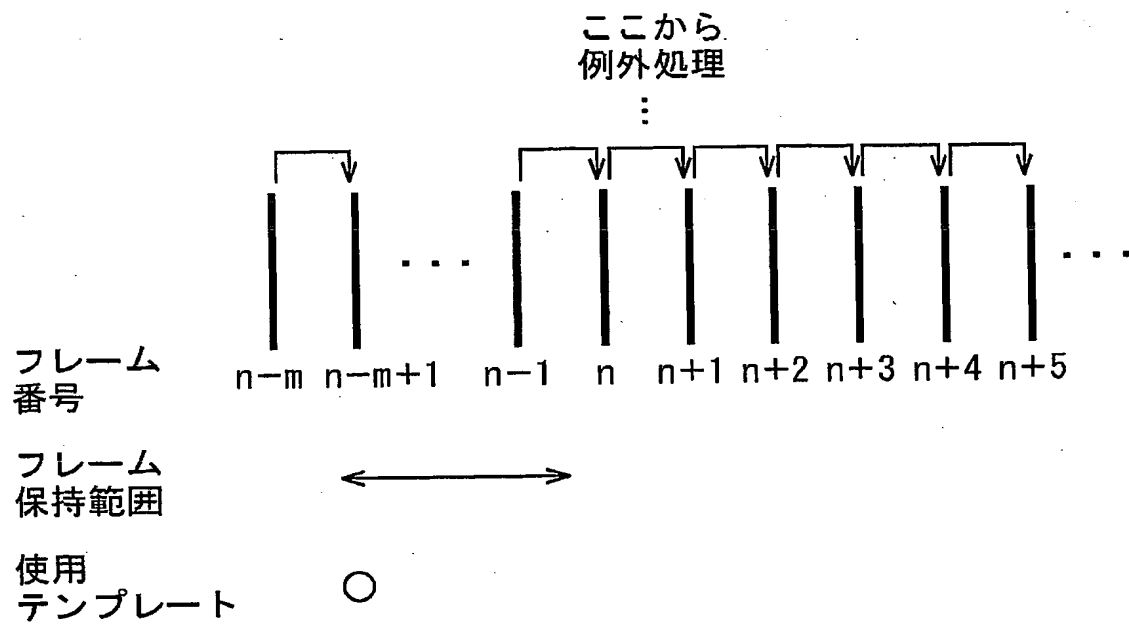


【図 35】  
図35

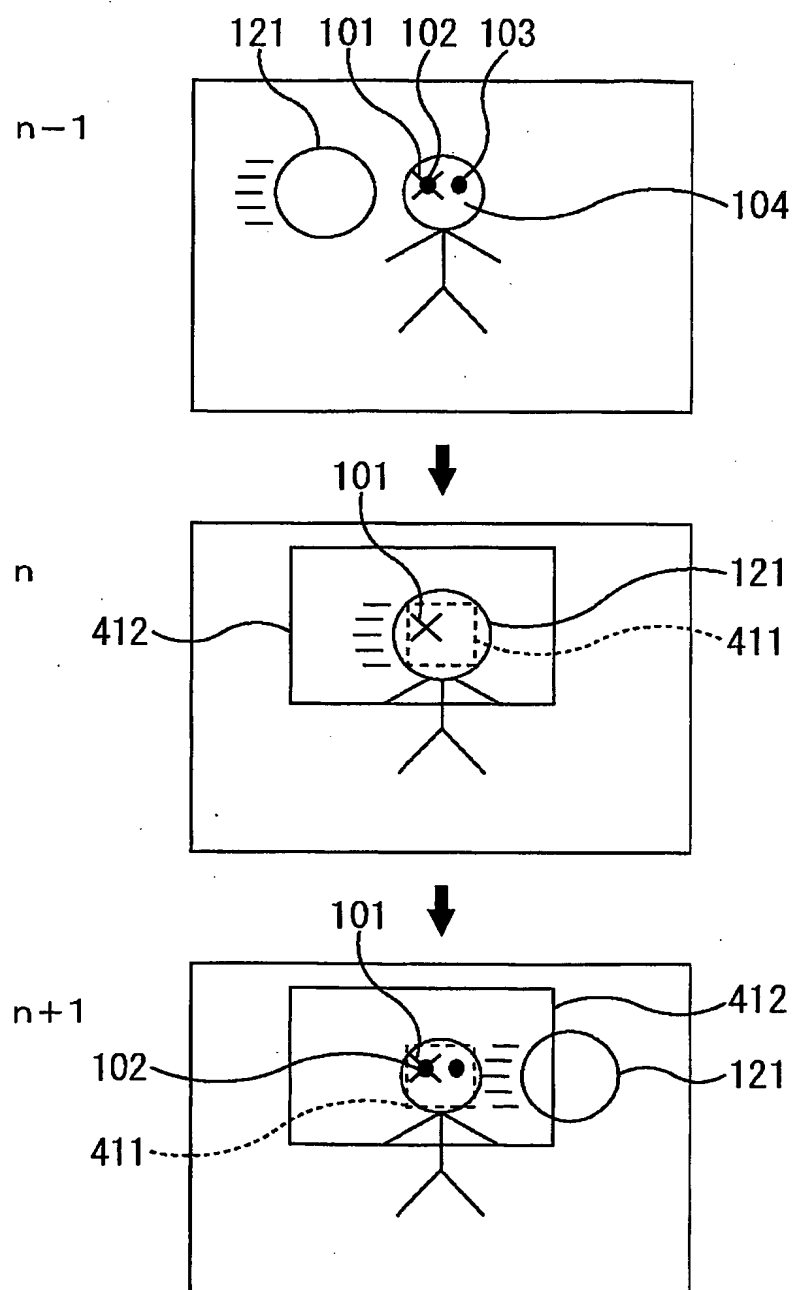
【図 36】  
図36



【図 37】  
図37



【図 38】  
図38





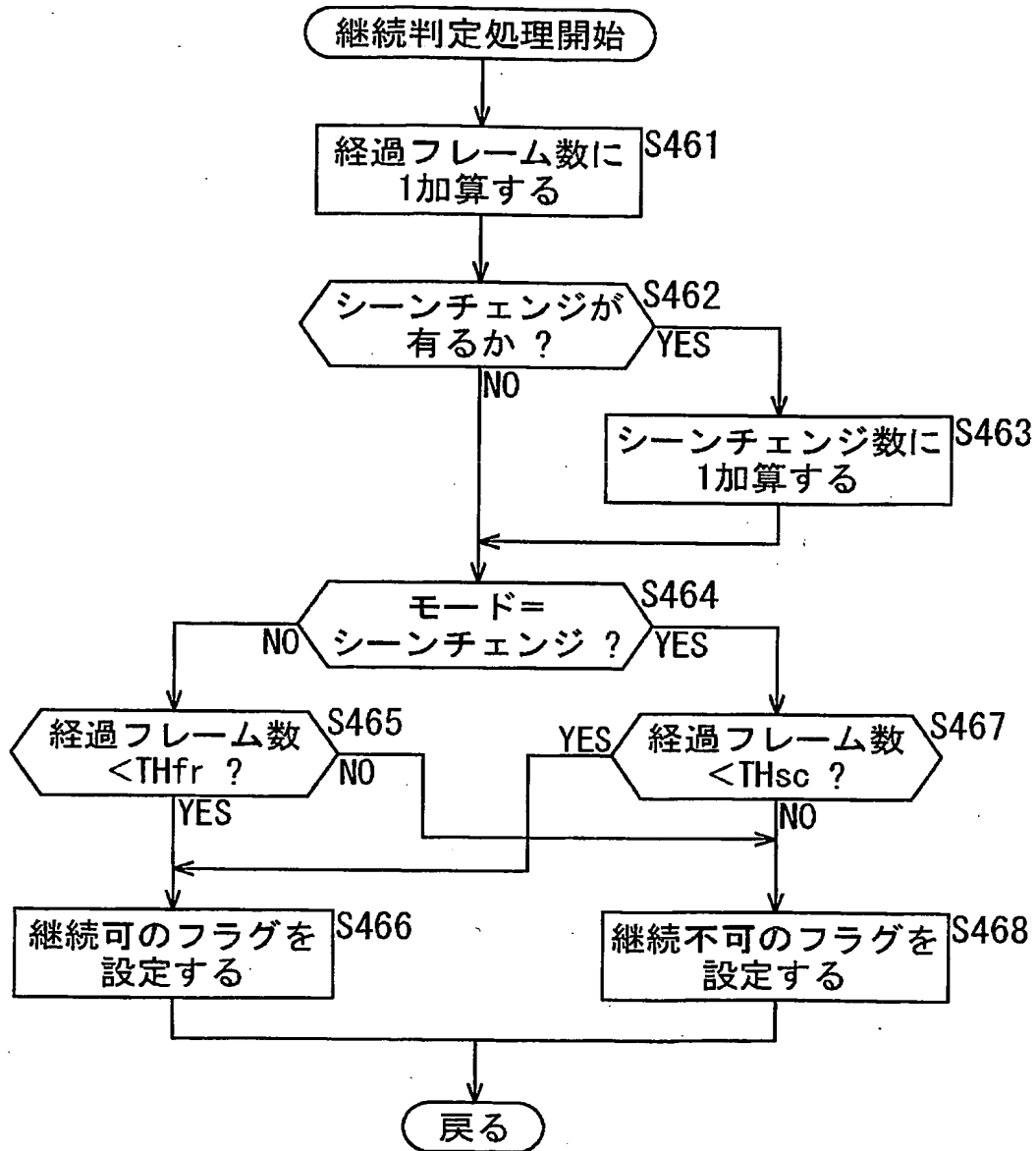
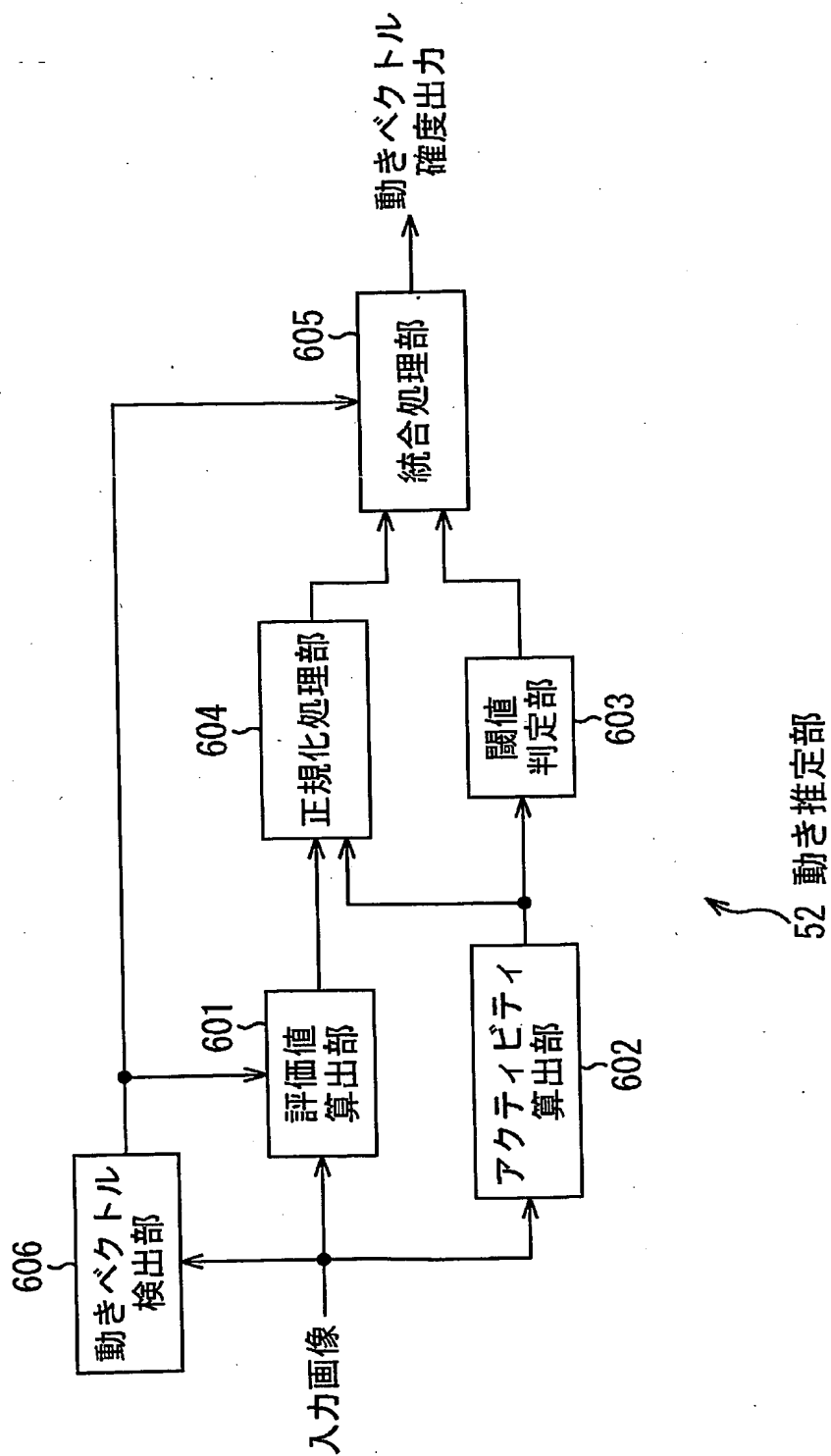
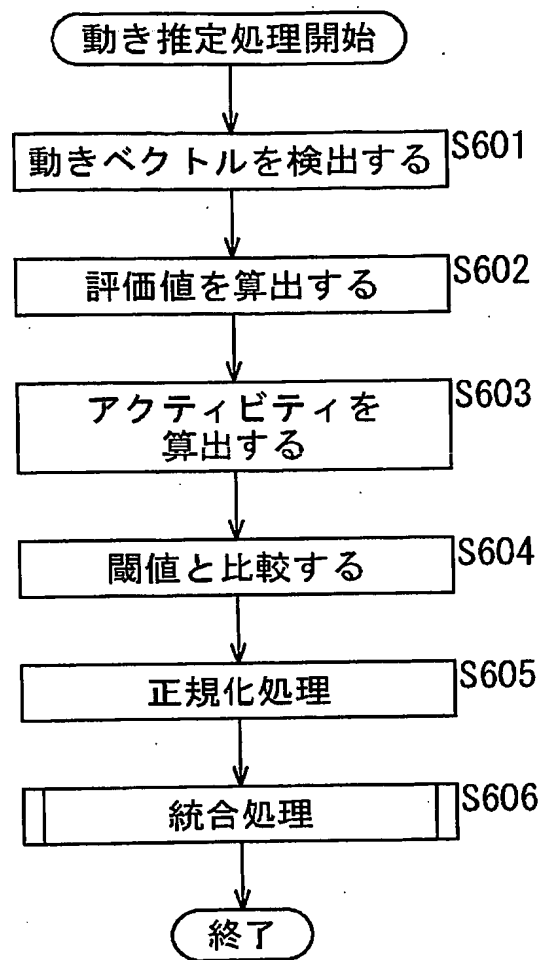
【図 39】  
図39

図40

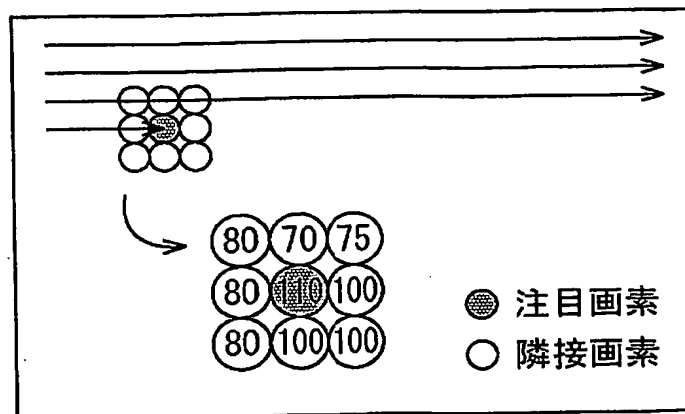
【図40】



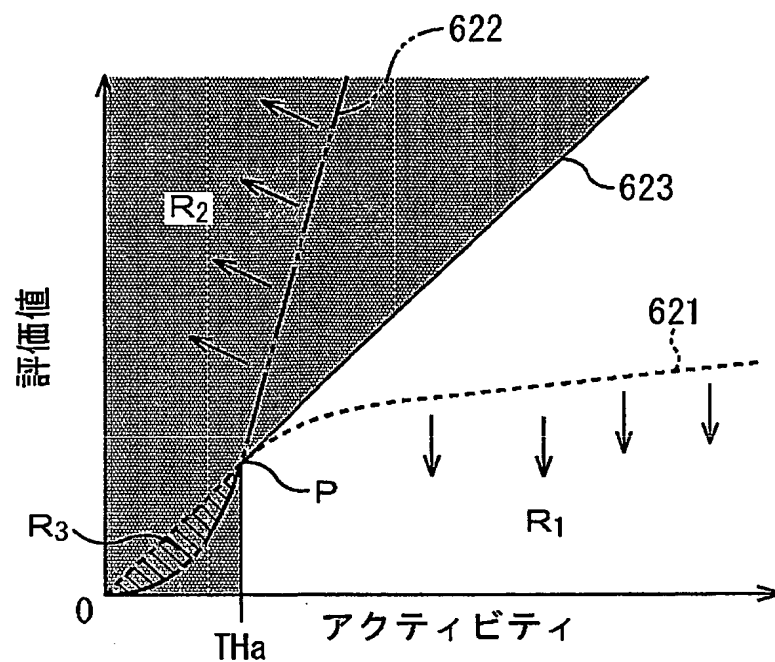
【図 4 1】  
図41



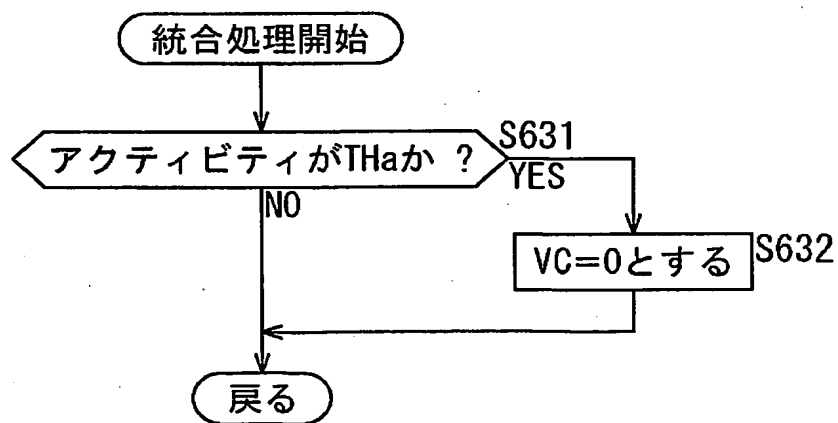
【図 4 2】  
図42



【図 4 3】  
図43

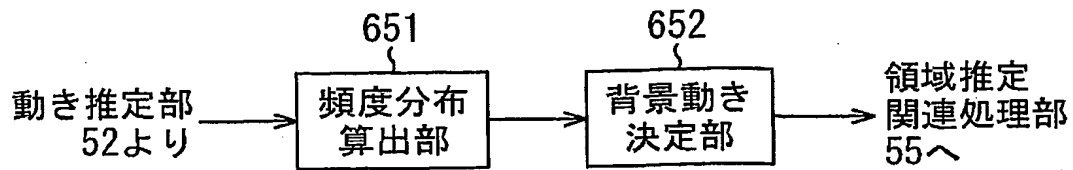


【図 4 4】  
図44



【図45】

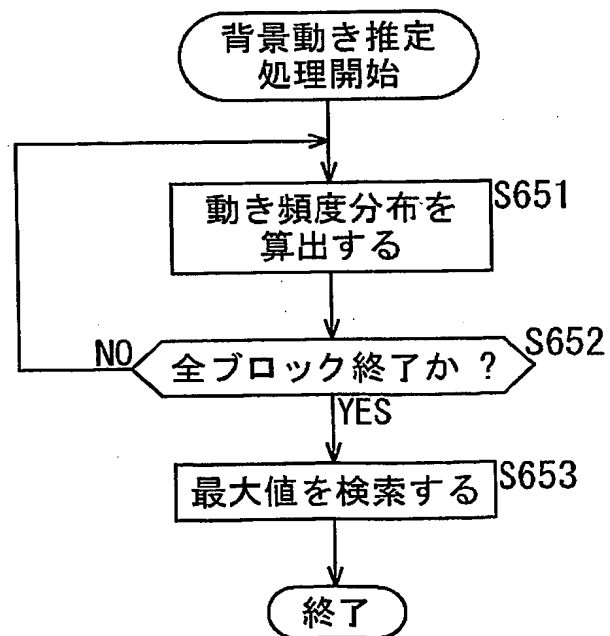
図45



54 背景動き推定部

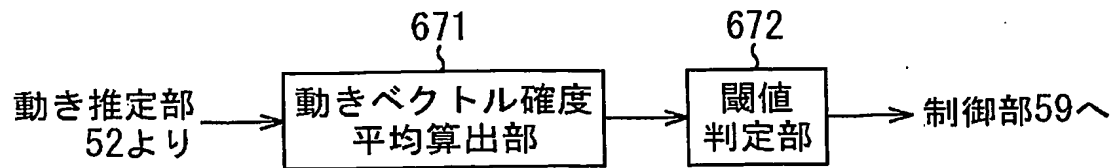
【図46】

図46



【図 47】

図47



53 シーンチェンジ検出部

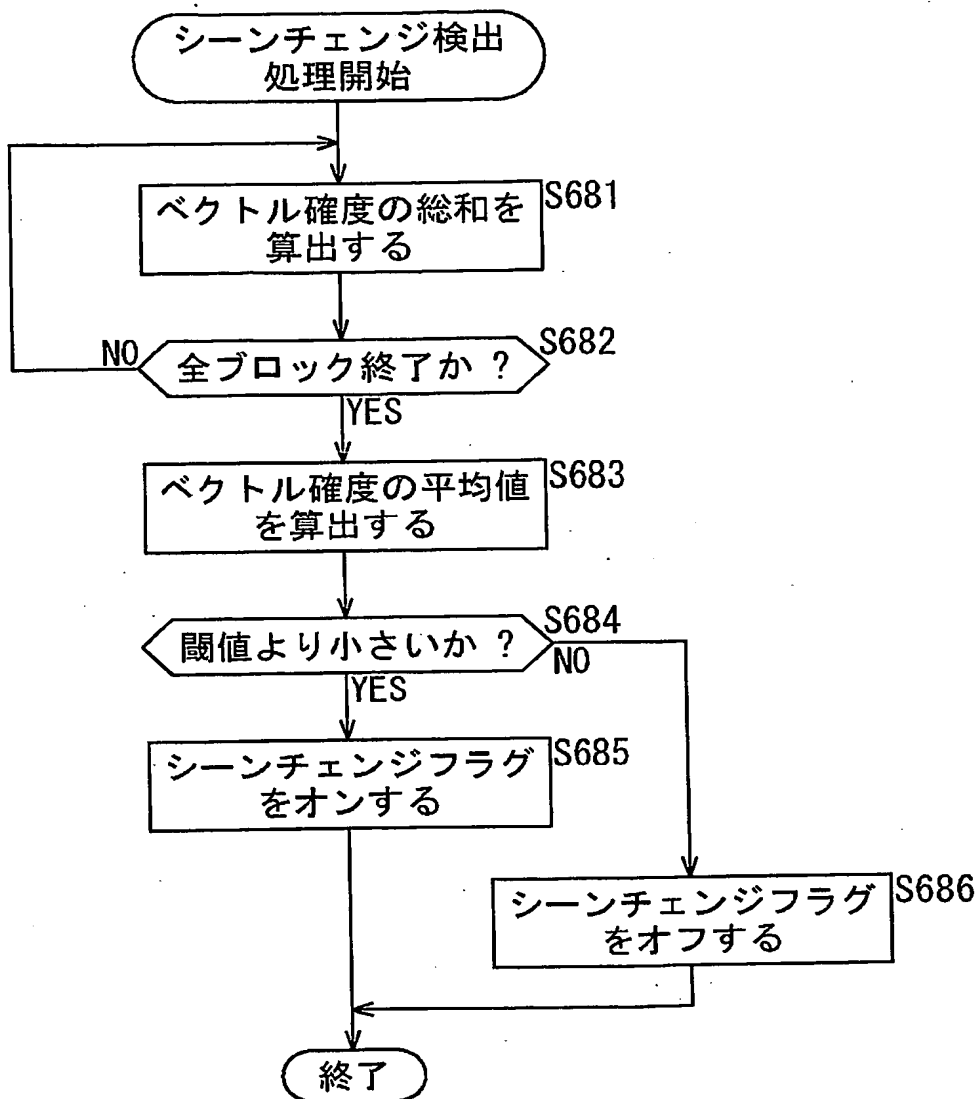
【図 48】  
図48

図49

【図49】

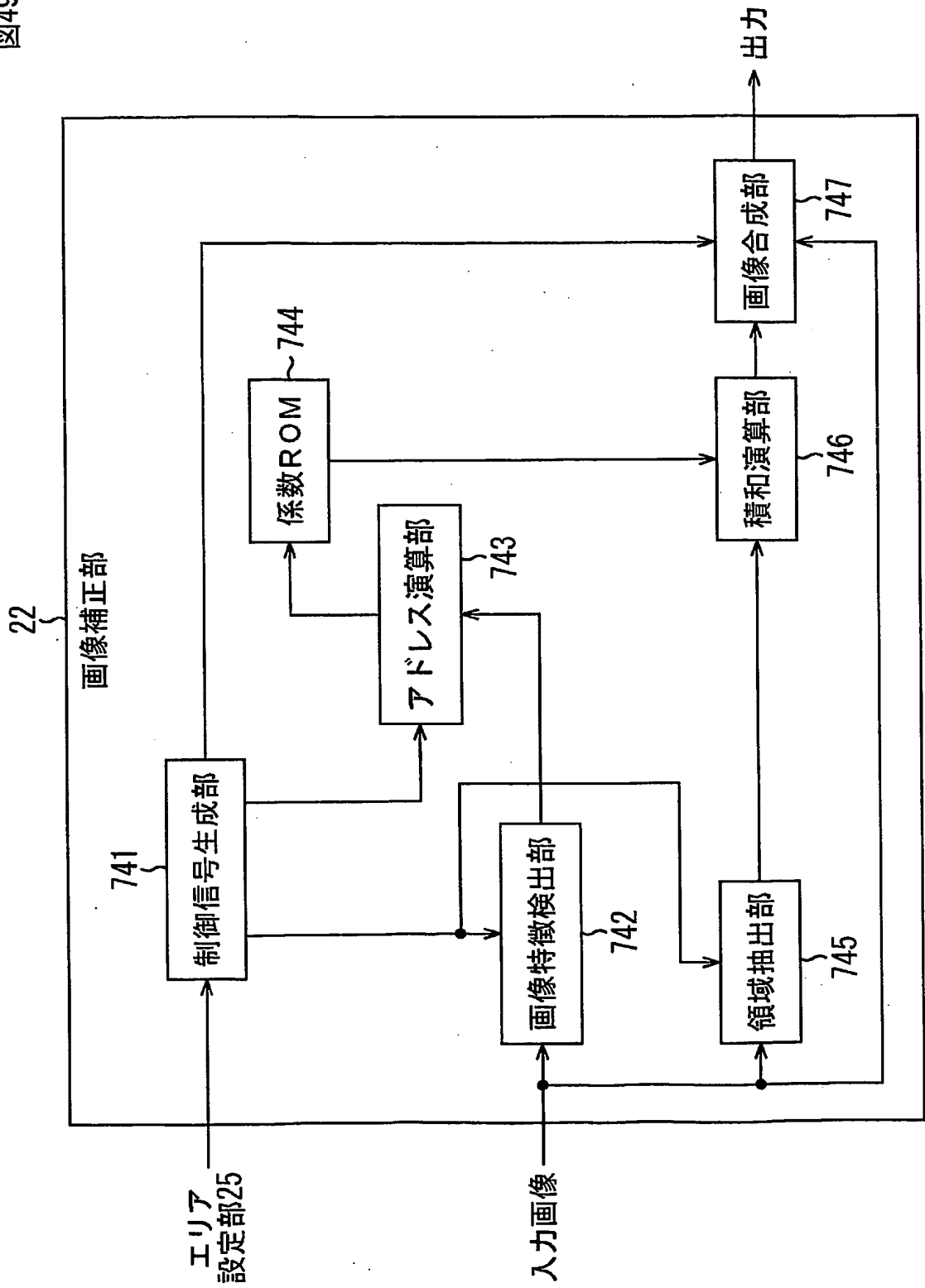




図50

【図50】

名称	関連するブロック	意味
制御信号 A	画像合成部、 領域抽出部	処理領域を特定する パルス
制御信号 B	アドレス演算部	ぼけの度合いを表す パラメータ $\sigma$
制御信号 C	アドレス演算部	関係式の重み $W_a$ の 切り替え
制御信号 D	画像特徴検出部	閾値の切り替え

【図 51】  
図51

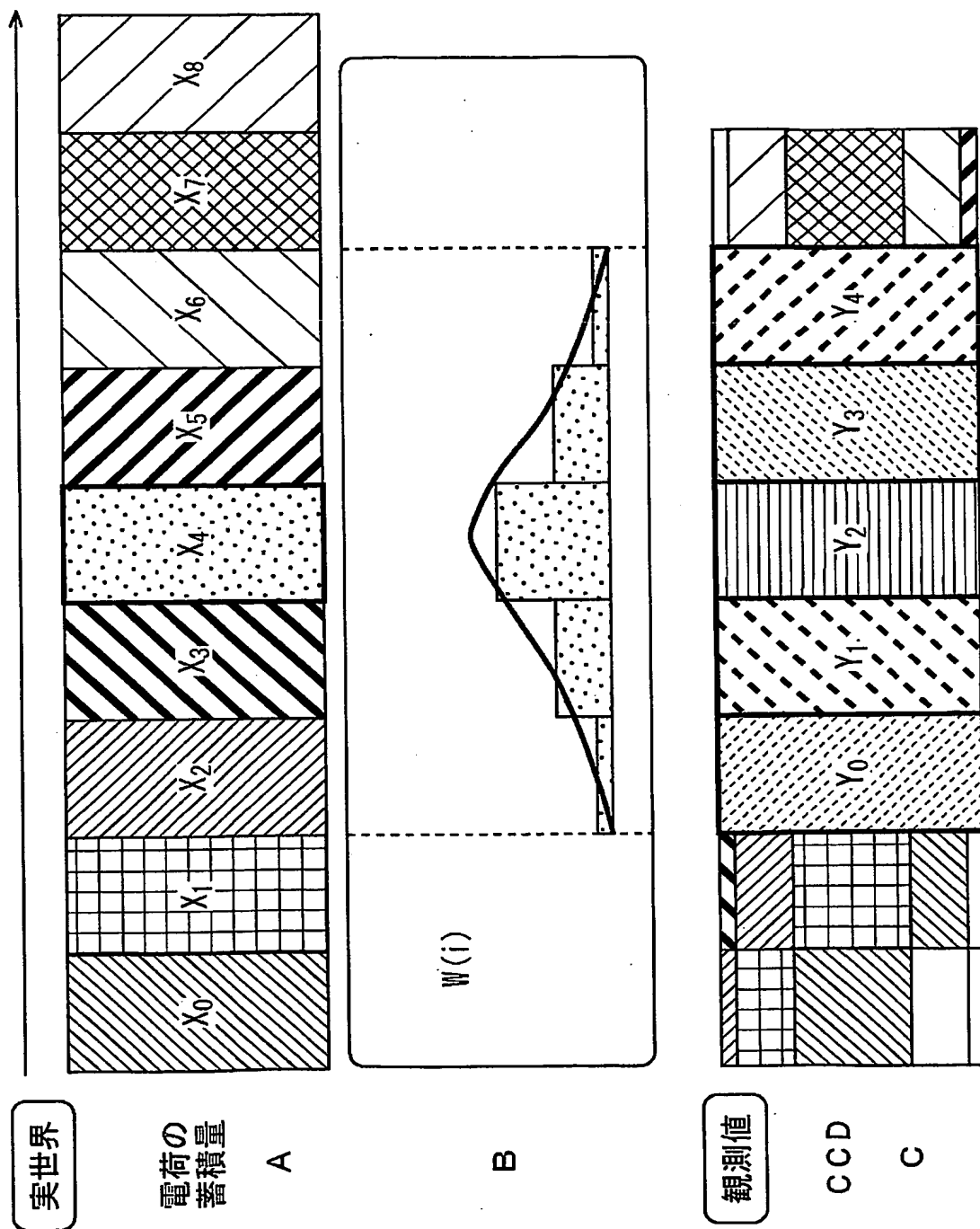
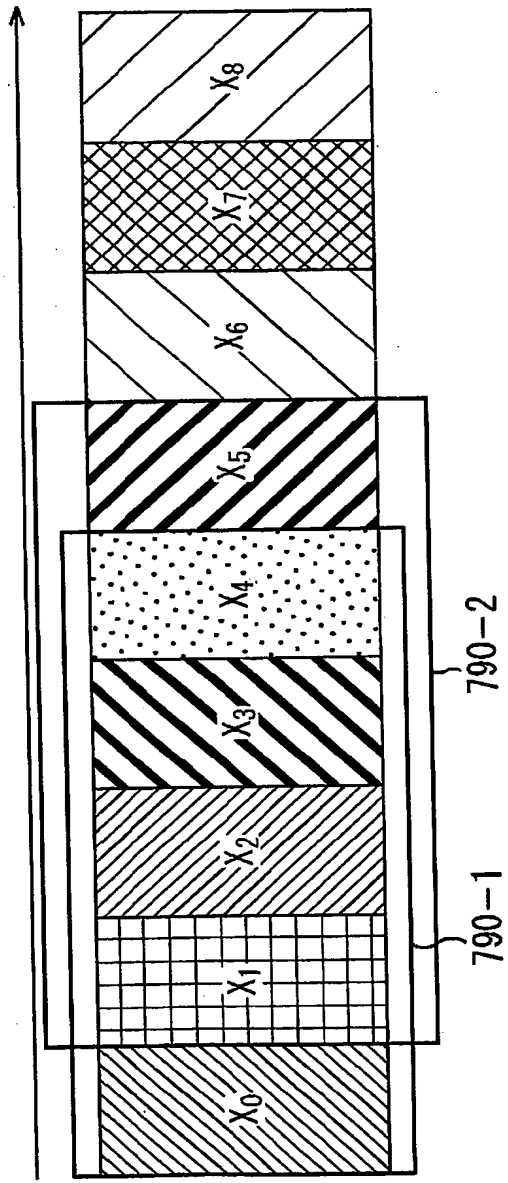


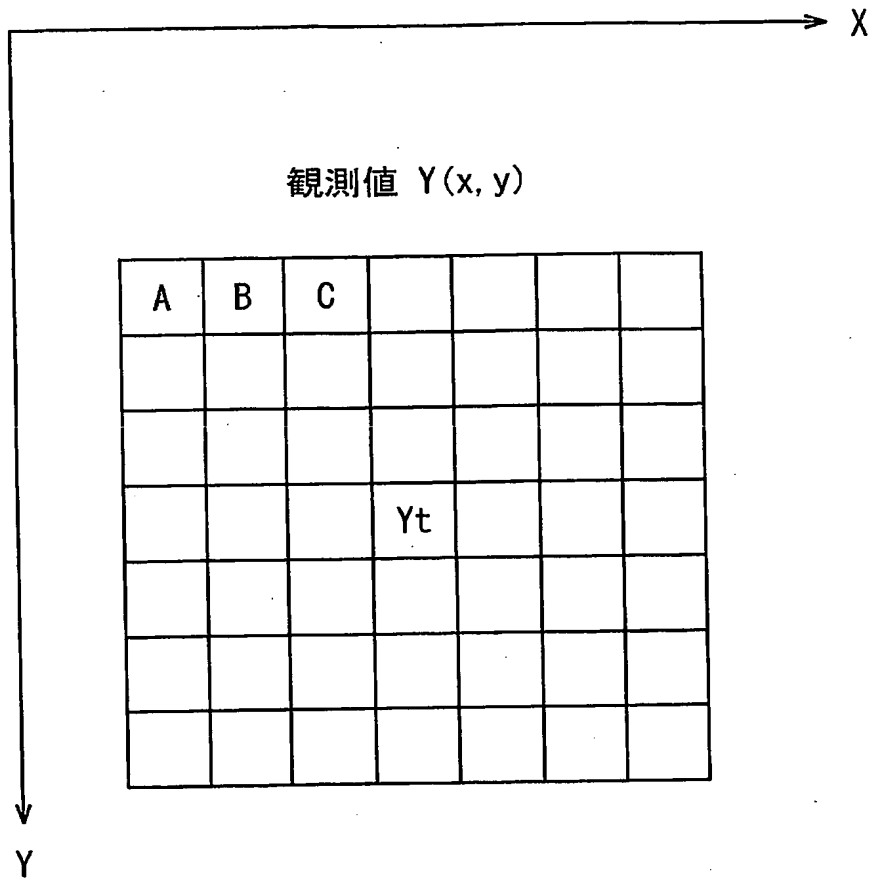
図52

【図 5 2】



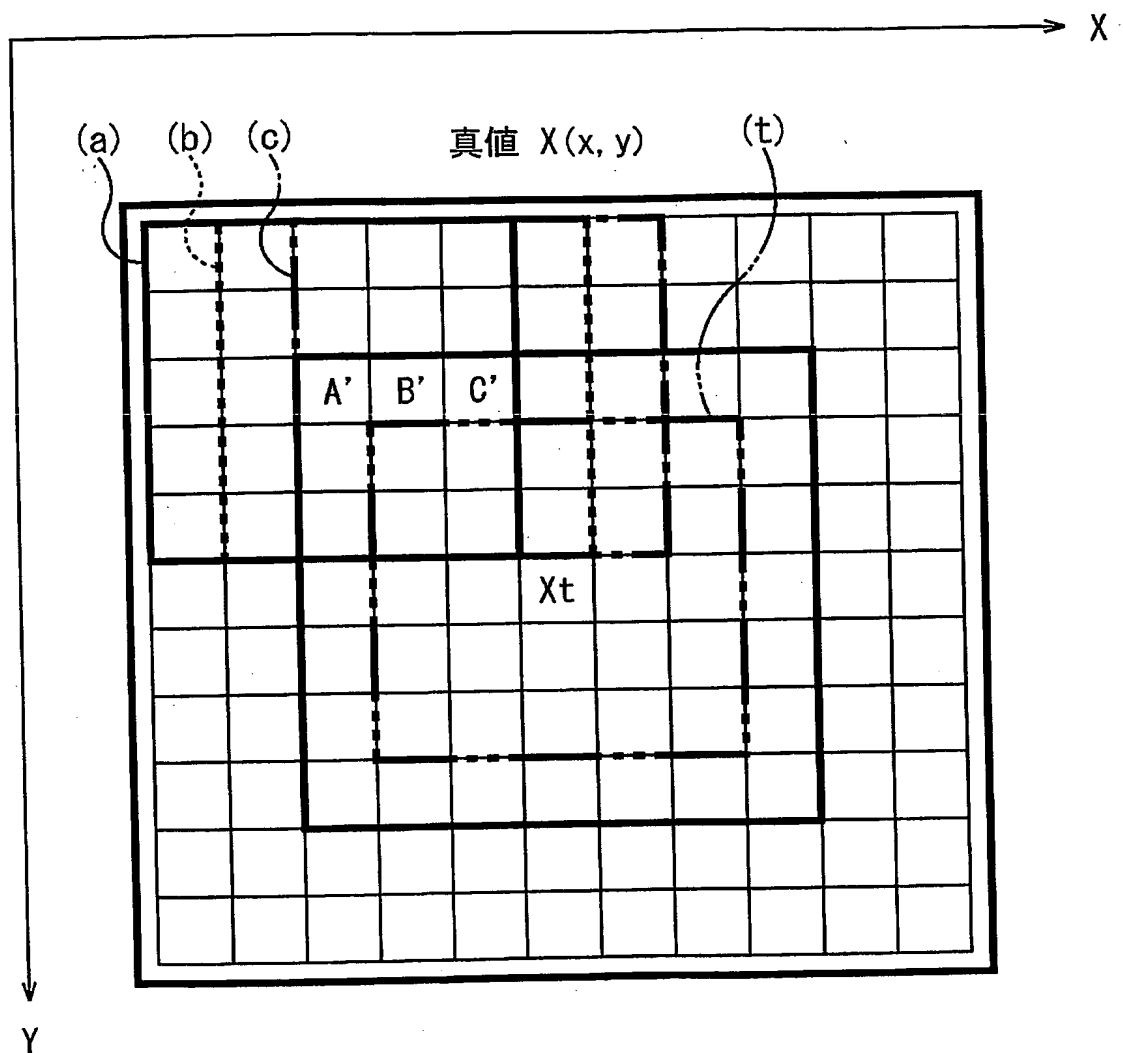
【図 53】

図53



【図 54】

図54



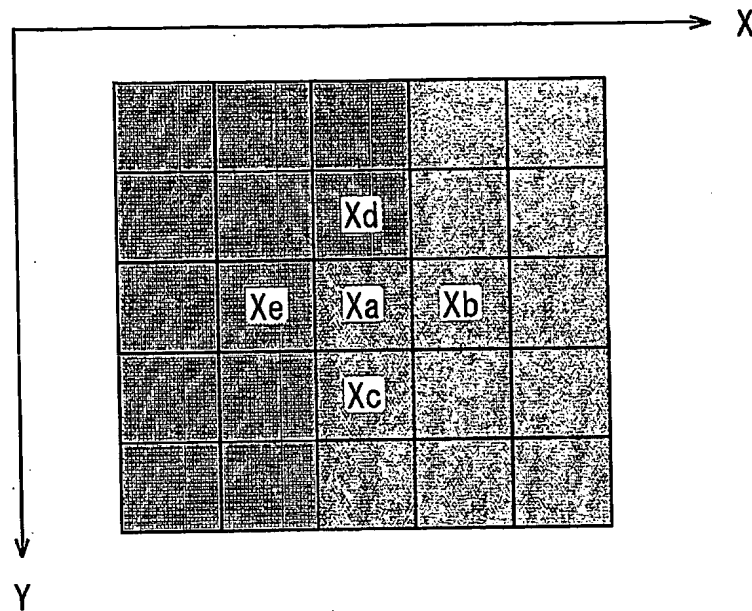
【図55】

図55

パラメータコードp <sub>2</sub>	W <sub>1</sub>	W <sub>2</sub>	W <sub>3</sub>	W <sub>4</sub>
0001	0.1	0.1	0.1	5.4
0010	0.1	0.1	5.4	0.1
0100	0.1	5.4	0.1	0.1
1000	5.4	0.1	0.1	0.1
0011	0.1	0.1	2.6	2.6
0101	0.1	2.6	0.1	2.6
0110	0.1	2.6	2.6	0.1
1001	2.6	0.1	0.1	2.6
1010	2.6	0.1	2.6	0.1
1100	2.6	2.6	0.1	0.1
0111	0.3	1.7	1.7	1.7
1011	1.7	0.3	1.7	1.7
1101	1.7	1.7	0.3	1.7
1110	1.7	1.7	1.7	0.3
1111	1.3	1.7	1.3	1.3

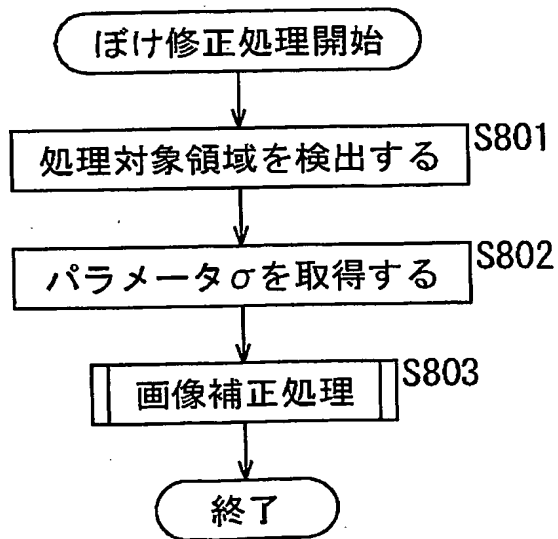
【図 56】

図56



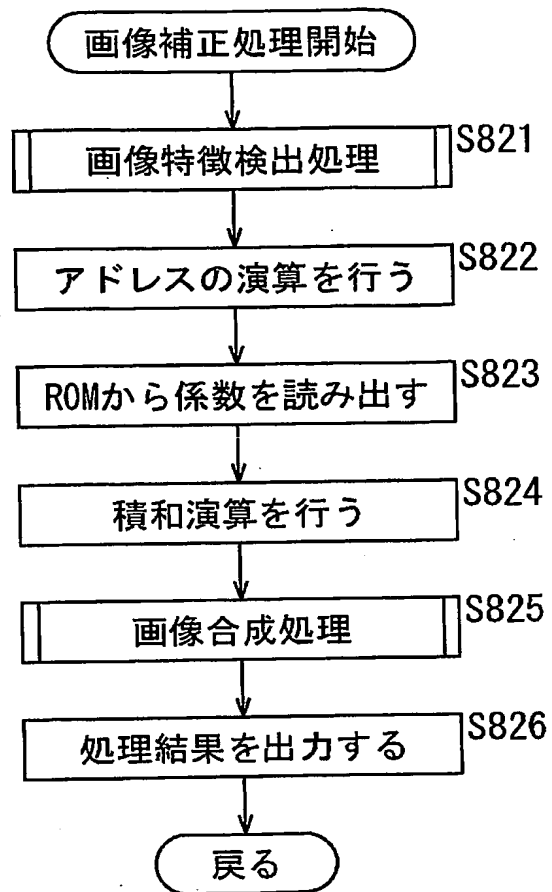
【図 57】

図57



【図 58】

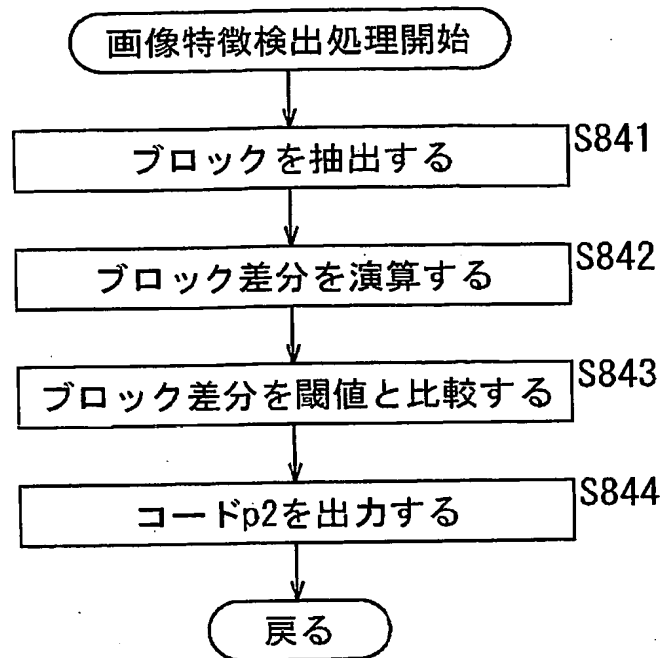
図58



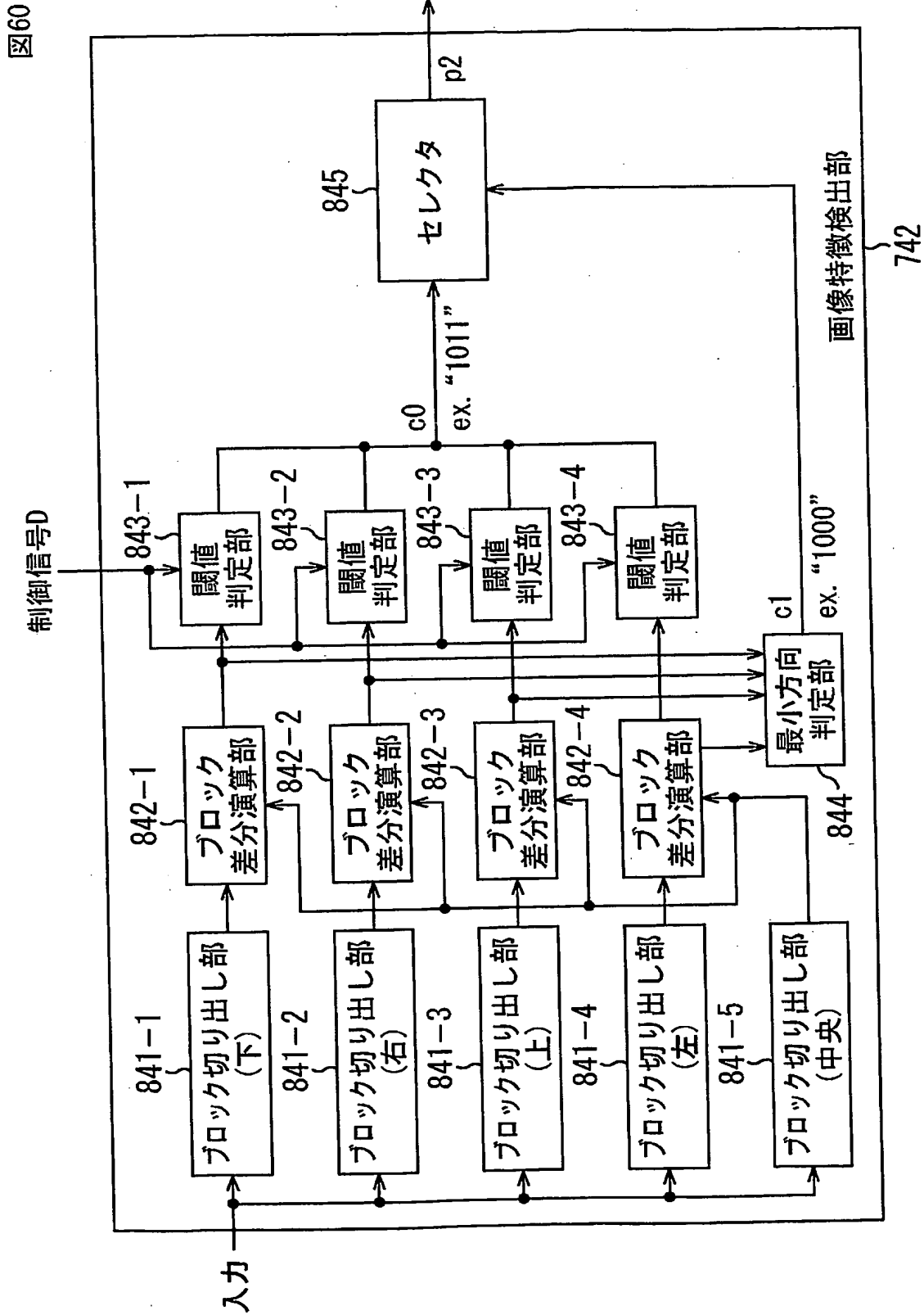


【図 59】

図59

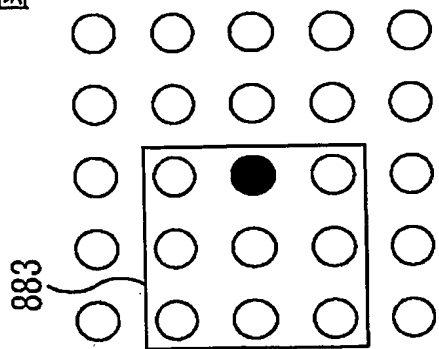


【図 60】

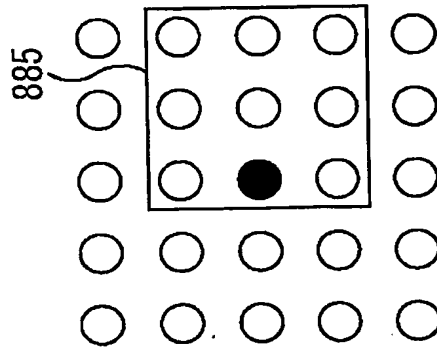


【図61】

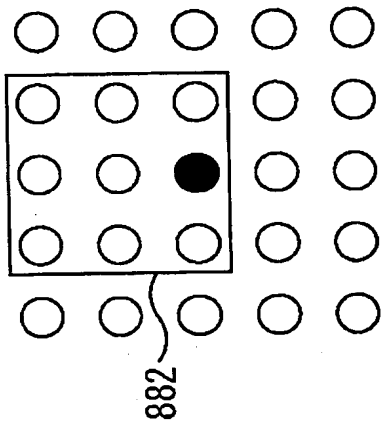
図61



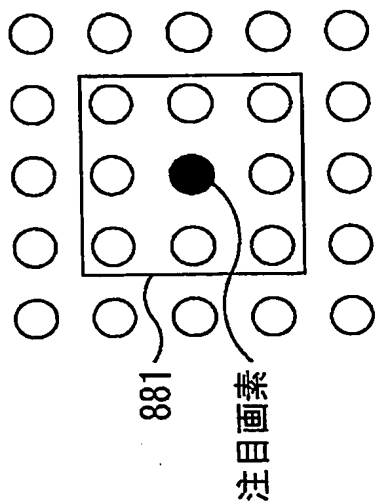
C



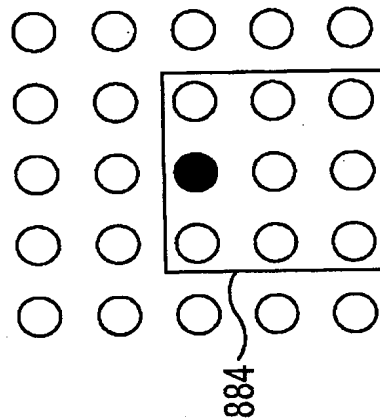
B



A

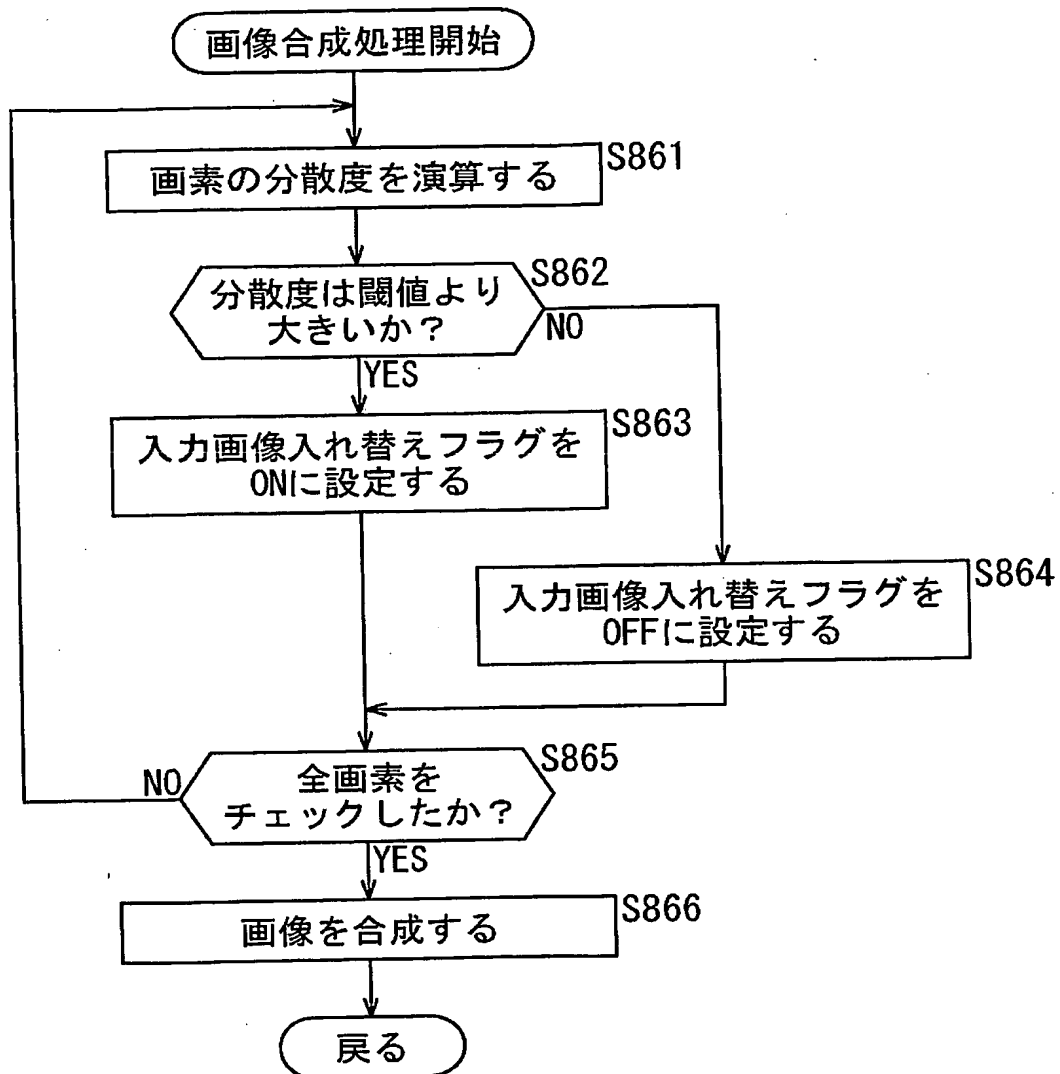


E



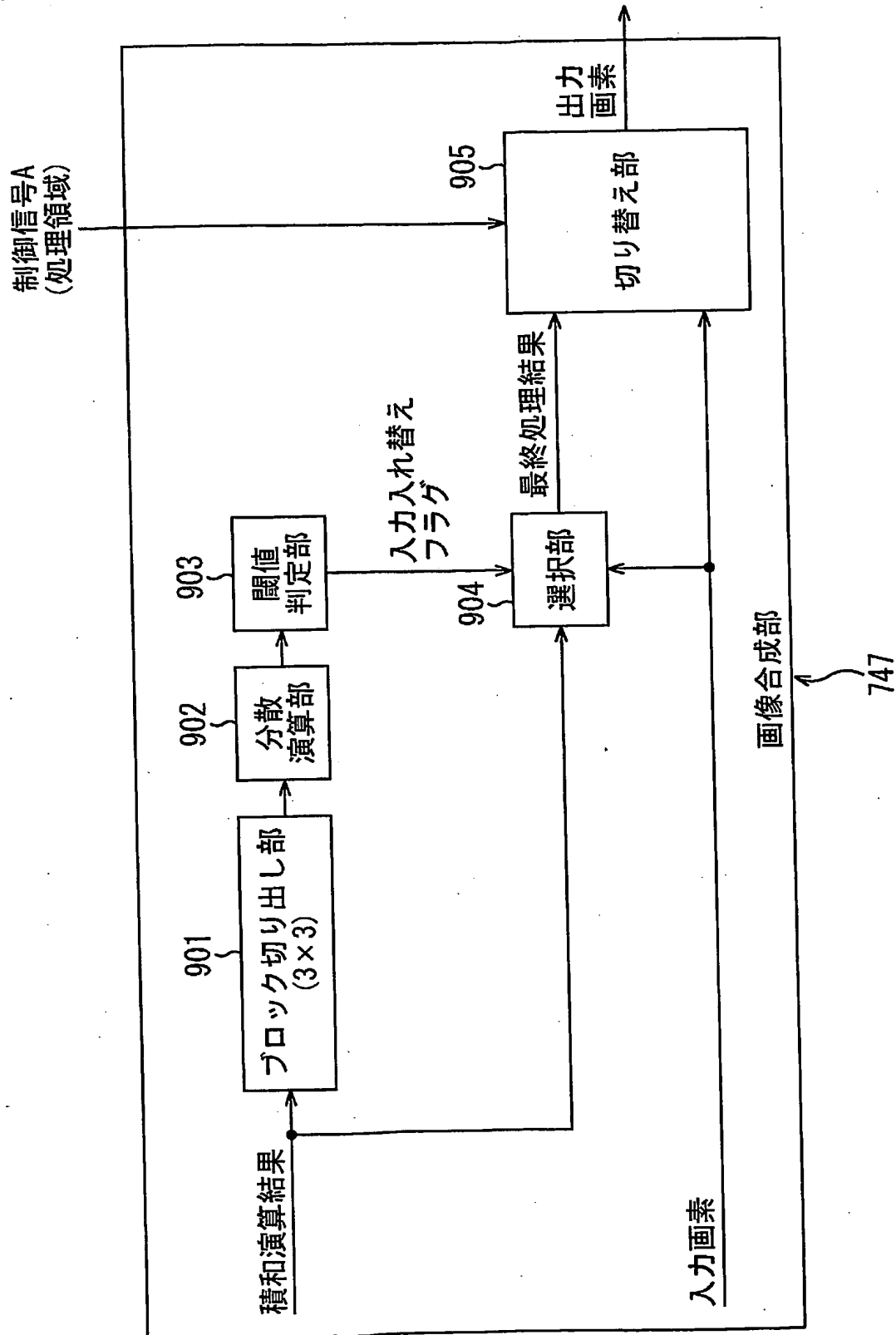
【図 62】

図62



【図 63】

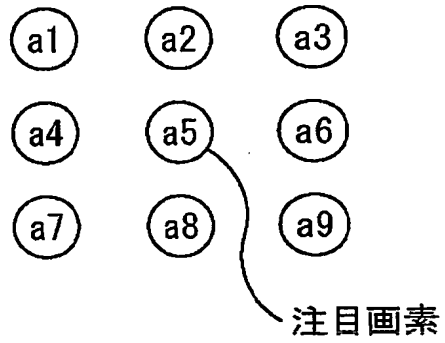
図 63





【図 64】  
図64

分散演算



**【書類名】要約書****【要約】**

**【課題】** 画像の中で移動するオブジェクトを発見しやすく表示できるようにする。

**【解決手段】** 撮像部 21 がオブジェクトの周囲の画像を撮像し、移動するオブジェクトの追尾点を追尾対象検出部 24 が検出する。オブジェクト追尾部 26 は、オブジェクトの追尾点を追尾し、追尾結果をエリア設定部 25 に出力する。エリア設定部 25 は、画像の中で、オブジェクトが含まれる所定の領域を補正対象エリアとして設定する。画像補正部 22 は、補正対象エリア内の画像だけ、画像のぼけを除去するように補正し、画像ディスプレイ 23 に出力する。画像ディスプレイ 23 は、注目すべきオブジェクトが鮮明であって、背景がぼけた画像を表示する。本発明は、監視カメラシステムに適用できる。

**【選択図】** 図 1

特願 2004-000752

出願人履歴情報

識別番号

[000002185]

1. 変更年月日

1990年 8月30日

[変更理由]

新規登録

住所

東京都品川区北品川6丁目7番35号

氏名

ソニー株式会社